



UNIVERSITY
OF
JOHANNESBURG

COPYRIGHT AND CITATION CONSIDERATIONS FOR THIS THESIS/ DISSERTATION



- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- NonCommercial — You may not use the material for commercial purposes.
- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

How to cite this thesis

Surname, Initial(s). (2012) Title of the thesis or dissertation. PhD. (Chemistry)/ M.Sc. (Physics)/ M.A. (Philosophy)/M.Com. (Finance) etc. [Unpublished]: [University of Johannesburg](https://ujdigispace.uj.ac.za). Retrieved from: <https://ujdigispace.uj.ac.za> (Accessed: Date).

'n ONTWERPSMETODOLOGIE
VIR
VERSPREIDE DATABASISSE.

deur
Anton Rossouw

VERHANDELING
voorgelê ter vervulling van vereistes vir die graad
MAGISTER IN DIE BESTUURSWETENSKAPPE
in
REKENAARWETENSKAP
in die
FAKULTEIT EKONOMIESE EN BESTUURSWETENSKAPPE
aan die
RANDSE AFRIKAANSE UNIVERSITEIT

1988

STUDIELEIER: Prof S.H. Von Solms

DOEL, MOTIVERING EN OPSOMMING

Doel

Die doel van hierdie verhandeling is om 'n raamwerk vir 'n metodologie voor te stel, wat spesifiek van toepassing is op die ontwerp van verspreide databasisse. Hierdeur word die vraag beantwoord: "Watter aspekte in die ontwerp van 'n verspreide databasis moet behandel word, sodat die voordele wat verspreiding inhou, maksimaal is?"

Motivering

Ons is daarvan oortuig dat die proses van verspreide databasisontwerp streng metodologies benader moet word om sodoende 'n effektiewe ontwerp tot gevolg te hê. Die spesiale struktuur van 'n verspreide databasis skep moontlike probleme wat met 'n metodologiese oplossing gehanteer kan word. In hierdie verhandeling poog ons nie om 'n spesifieke metodologie in besonder voor te stel nie, maar om wel 'n moontlike raamwerk vir 'n gedetailleerde metodologie te vorm. Hierdie raamwerk sal grootliks gebaseer word op werk wat reeds binne die veld van die gesentraliseerde databasis gedoen is, omdat daar tans min navorsing in terme van verspreide databasisontwerp bestaan.

Opsomming

As primêre uitgangspunt vir 'n metodologiese raamwerk, kombineer ons twee moderne en besproke datamodelle nl. die entiteitsverwantskapmodel en relasiemodel. 'n Stapsgewyse metodologiese raamwerk word ontwikkel deur verskillende fases van ontwerp voor te stel. Die voordele wat elke fase vir verspreide databasisontwerp inhou, sal deurgaans uitgelig word. Dit moet egter beklemtoon word dat hierdie metodologiese raamwerk net as een van talle moontlike benaderings beskou moet word.

I N H O U D S O P G A W E

ONDERWERP

BLADSY

HOOFSTUK 1:	DIE ONTWIKKELING VAN 'N ONTWERPSMETODIEK, DEUR DIE ANALISE VAN BETROKKE ASPEKTE BY DATABASISSE, VERSPREIDING EN DATAMODELLE	
1.1	INLEIDING.	1
1.2	DIE ROL VAN VERSPREIDE INLIGTING BINNE DIE ORGANISASIE.	4
1.3	GESENTRALISEERDE ONTWERP TEENoor VERSPREIDE ONTWERP....	6
1.4	HOEKOM IS VERSPREIDE DATABASISSE BELANGRIK ?	8
1.4.1	DIE STRUKTUUR EN GEOGRAFIESE VERSPREIDING VAN DIE ORGANISASIE	9
1.4.2	KOSTEBESPARINGS.	10
1.4.3	DIE AANPASBAARHEID VAN VERSPREIDE DATABASISSE TOT VERANDERING.	12
1.5	DIE ONTWIKKELING VAN 'N VERSPREIDE DATABASIS ONTWERPSMETODOLOGIE	13
1.5.1	'n ONTLEDING VAN TRADISIONELE ONTWERPFASES	14
1.5.2	ASPEKTE BETROKKE BY DIE ONTWERP VAN VERSPREIDE DATABASISSE.	16
1.5.3	VEREENVOUDIGDE ONTWERPFASES VIR VERSPREIDE-DATABASISSE	22
1.5.4	DIE ROL VAN DATAMODELLE IN DIE ONTWERPSPROSES	26
1.5.4.1	DATAMODEL KLASSE EN VOORSTELLINGS	27
1.5.5	ONTWERPSTRATEGIË VIR VERSPREIDE DATABASISSE	29
1.5.5.1	BO-NA-ONDER ONTWERP	30
1.5.5.2	ONDER-NA-BO ONTWERP	31
1.5.5.3	KOMBINASIE VAN BO-NA-ONDER EN ONDER-NA-BO BENADERINGS	31
1.5.6	DIE KEUSE VAN DATAMODELLE VIR VERSPREIDE DATABASISSE	32
1.5.6.1	DIE ENTITEITVERWANTSKAPMODEL	36
1.5.6.2	DIE RELASIEMODEL	38

1.5.7	FINALISERING VAN 'n METODOLOGIESE RAAMWERK VIR VERSPREIDE DATABASISONTWERP.....	42
1.5.7.1	SUKSES-FAKTORE BETROKKE BY 'n ONTWERPSMETODOLOGIE.....	42
1.5.7.2	DIE SINTESE VAN 'n METODOLOGIESE RAAMWERK VIR VERSPREIDE DATABASISONTWERP.....	44
1.6	DIE STRUKTUUR VAN DIE VERHANDELING.....	47
1.7	OPSOMMING.....	50

**HOOFSTUK 2 : FASE I : ORGANISASIE ANALISE - BEHOEFTEBEPALING EN
ONTLEDING**

2.1	INLEIDING.....	52
2.2	ONTLEDINGSFASES.....	54
2.3	DIE VERSPREIDE DATABASIS SE INPAK OP INLIGTINGVLOEI....	55
2.4	INLIGTINGSBRONNE BINNE DIE VERSPREIDE ORGANISASIE.....	57
2.5	DATA-ANALISE.....	59
2.5.1	FUNKSIONELE-ANALISE.....	60
2.5.2	ENTITEIT-ANALISE.....	61
2.5.3	MATRIKS-VOORSTELLING.....	62
2.6	ONTLEDING VAN DIE HUIDIGE VERSPREIDE ORGANISASIE.....	63
2.7	VERBETERING VAN DIE VERSPREIDE ORGANISASIE EN 'n TOEKOMSBLIK.....	65
2.8	OPSOMMING.....	67

HOOFSTUK 3: FASE II - VERSPREIDINGSANALISE

3.1	INLEIDING.....	69
3.2	PLEKONTLEDING VIR VERSPREIDE DATABASISONTWERP.....	71
3.3	AFFINITEITSANALISE.....	73
3.3.1	SAAMGROEPERING GEBASEER OP AFFINITEIT.....	75
3.4	OPSOMMING.....	77

HOOFSTUK 4 : FASE III : KONSEPTUELE-LOGIESE ENTITEITSVERWANTSKAP-MODELLERING

4.1	INLEIDING.....	79
4.2	'n KONSEPTUELE-LOGIESE MODEL VIR VERSPREIDE DATABASISSE.....	80
4.2.1	DIE EV-BENADERING EN DIE OPSTEL VAN KONSEPTUELE SKEMAS VIR VERSPREIDE DATABASISSE.....	81
4.3	VOORDELE VAN DIE EV MODEL VIR VERSPREIDE DATABASISONTWERP.....	83
4.4	VERSPREIDE DATABASIS GERIGTE EV ONTWERPSBENADERING.....	86
4.5	DIE ENTITEITSVERWANTSKAPMODEL.....	87
4.5.1	ENTITEITE, ATTRIBUTES EN VERWANTSKAPPE.....	89
4.5.2	KARDINALITEITE.....	90
4.5.4	SWAK ENTITEITE.....	91
4.6	'n UITGEBREIDE ENTITEITSVERWANTSKAPMODEL VIR VERSPREIDE DATABASISSE.....	91
4.6.1	ENTITEITE, ATTRIBUTES EN VERWANTSKAPPE.....	92
4.6.2	OBJEK-KLASSE.....	93
4.6.3	VERWANTSKAPGRAAD.....	94
4.6.4	VERBINTENISGRAAD.....	95
4.6.5	LIDMAATSKAPKLASSE.....	95
4.7	DIE TOEPASSING VAN DIE EV MODEL.....	96
4.8	'n UITGEBREIDE EV MODEL VIR VERSPREIDE ORGANISASIES.....	99
4.9	SINTESE VAN 'n VERSPREIDE EV MODEL.....	101
4.10	OMSKAKELING VANAF 'n EV MODEL NA 'n RELASIE MODEL.....	105
4.11	OPSOMMING.....	109

HOOFSTUK 5 : FASE IV : LOGIESE TOEPASSING SPESIFISERING:

5.1	INLEIDING.....	111
5.2	LOGIESE TOEPASSING SPESIFIKASIE VIR VERSPREIDE DATABASISONTWERP.....	113

5.3 VERDERE FUNKSIONELE-ANALISE OP VERSPREIDE DATABASISSE..... 115

5.3.1 FUNKSIONELE-ANALISE VAN VERSPREIDE PROSESSE..... 116

5.3.2 ENTITEITSVERWANTSKAP GEBASEERDE TALE VIR SPESIFISERING VAN VERSPREIDE DATABASIS TOEPASSINGS. 119

5.3.3 TOEPASSINGSGERIGTE AFFINITEITSANALISE..... 123

5.4 DIE ONTWERP VAN VERSPREIDE TRANSAKSIEMODELLE..... 124

5.5 OPSOMMING..... 125

HOOFSTUK 6 : FASE V : LOGIES - FISIESE RELASIEMODELLERING:

6.1 INLEIDING..... 128

6.2 'n LOGIES-FISIESE MODEL VIR VERSPREIDE DATABASISSE..... 129

6.3 DIE GEBRUIK VAN DIE RELASIEMODEL VIR VERSPREIDE DATABASISSE..... 130

6.3.1 BESTAAN DAAR SUIWER RELASIEMODELLE?..... 132

6.4 'n RAAMWERK VIR VERSPREIDE LOGIES-FISIESE RELASIEDATABASISONTWERP..... 135

6.5 'n BESPREKING VAN DIE RELASIEMODEL..... 137

6.5.1 ATTRIBUTES..... 138

6.5.2 RELASIESLEUTELS..... 139

6.5.3 RELASIEDINAMIKA EN RELASIEBETEKENIS..... 140

6.5.4 MANIPULASIE EN DEFINIEERING VAN RELASIESTRUKTURE.. 141

6.5.4.1 DIE SQL TAAL EN VERSPREIDE DATABASISSE..... 142

6.5.4.2 DIE STRUKTUUR VAN SQL..... 144

6.6 DOELWITTE VIR DIE ONTWERP VAN VERSPREIDE RELASIE DATABASISSE..... 146

6.7 OORSKAKELING VANAF DIE EV-MODEL NA 'n VERSPREIDE RELASIEMODEL..... 148

6.7.1 DIE VERSPREIDE EN GESENTRALISEERDE RELASIEMODEL... 150

6.8 DIE ONTWERP VAN DATABASIS VERSPREIDING..... 152

6.8.1 DATABASIS-FRAGMENTASIE..... 153

6.8.2 DIE ONTWERP VAN FRAGMENTASIE..... 154

6.8.2.1 HORISONTALE FRAGMENTASIEONTWERP..... 154

6.8.2.2	VERTIKALE FRAGMENTASIEONTWERP.....	157
6.8.2.3	KOMBINASIE VAN VERTIKALE EN HORIZONTALE FRAGMENTASIE.....	159
6.8.3	DIE TOEWYSING VAN FRAGMENTE.....	159
6.9	OPSOMMING.....	161

**HOOFSTUK 7: SLOTSOM - 'n METODOLOGIE VIR VERSPREIDE
DATABASISONTWERP**

7.1	INLEIDING.....	163
7.2	SAMEVATTING VAN DIE VOORGESTELDE METODOLOGIESE RAAMWERK.....	164
7.3	SLOTSOM.....	167
	BRONNELYS.....	169

HOOFSTUK 1

DIE ONTWIKKELING VAN 'N ONTWERPSMETODIEK, DEUR DIE ANALISE VAN BETROKKE ASPEKTE BY DATABASISSE, VERSPREIDING EN DATAMODELLE:

Opsomming

Hierdie hoofstuk is 'n oorsig van die onderwerpe, wat in die verhandeling behandel word. Ons kyk ook na die redes vir regverdiging van verspreide databasisse, en die sintese van 'n moontlike raamwerk vir 'n ontwerpsmetodiek vir verspreide databasisse. Dit sal bestaan uit 'n kombinasie van ontwerpsfases, ontwerpsbenaderings en toepaslike datamodelle. Die verspreidingsgedagte sal ook deurgaans, in die bespreking, uitgelig word. Die laaste gedeelte van hierdie hoofstuk verskaf 'n uitleg van die ander hoofstukke in die verhandeling. Ons vorm en bespreek 'n moontlike metodologie wat bestaan uit fases vir ontleding, verspreidingsanalise, logiese entiteitverwantskapmodellering, toepassing spesifisering en logies-fisiese relasiemodellering wat spesifiek vir verspreide databasisontwerp gebruik kan word.

1.1 INLEIDING

In hierdie verhandeling word die klem geplaas op die gebruik van datamodelle in 'n metodologiese benadering vir die ontwerp van 'n verspreide databasis. Daar word twee spesifieke datamodelle gebruik, nl. die entiteitverwantskap- en relasiemodelle. Ons poog om te bewys dat die gebruik van hierdie twee datamodelle binne 'n wel-deurdagte metodologie, die effektiewe ontwerp van 'n verspreide databasis tot gevolg sal hê. Hierdeur toon ons dat die ontwerp van 'n verspreide databasis gedeeltelik ooreenstem met die ontwerp van 'n gesentraliseerde databasis, alhoewel daar spesiale oorwegings

by elk is. Dit moet onthou word dat die klem in al die hoofstukke, op verspreidingskonsepte geplaas gaan word. Ons wil nie 'n gesentraliseerde ontwerpmetodologie omskakel in 'n verspreide ontwerpmetodologie nie, maar vroegtydig reeds ons aandag spits op die verspreidingsprobleem. 'n Presiese en doeltreffende metodologie is ook 'n ideale kandidaat vir outomatisasie (CASE - Computer Aided Software Engineering).

Die invloed wat verspreiding op databasisontwerp het, moet sterk beklemtoon word. Die ontstaan van databasisteorie is te danke aan die mens se strewe om inligting te verwerk. Dit word omvat deur die mens se strewe om voort te bou, en om data te versprei. Die mensdom gebruik masjiene as hulpmiddels om hulle doel te bereik, en dit sluit die gebruik van rekenaars en databasisse in. Ons kan met goeie databasisontwerp, die mens/masjien interaksie optimaliseer, en menslike behoeftes bevredig. Die doel van hierdie verhandeling is om 'n raamwerk voor te stel waarin 'n doeltreffende ontwerpmetodologie ontwikkel kan word, sodat die probleme van verspreide databasisontwerp geminimaliseer kan word.

Let wel dat daar nie 'n gedetailleerde metodologie ontwikkel word nie, maar wel net 'n basiese raamwerk vir verdere ontwikkeling van die metodologie. Die verhandeling poog nie om die perfekte oplossing te vind nie, maar wel om 'n metodologiese raamwerk te verskaf waarin die leser se gedagtes gestruktureer kan word.

In terme van verspreide databasisse noem Oxborrow [38] (p190) 'n paar belangrike vrae waaraan aandag geskenk moet word, wanneer die implementering van verspreide databasisse oorweeg word:

1. Is verspreide databasisse ekonomies en tegnies uitvoerbaar, en uit watter oogpunt moet dit aangepak word? (in teenstelling met gesentraliseerde databasisse).

2. Met watter argitektuur moet die kommunikasienetwerk geïmplementeer word en watter databasisbedryfstelsel (DBBS) is van toepassing ?
3. Hoe moet die toepassings ontwerp word om die verspreide struktuur voor te stel, en sal die fisiese databasisnodes heterogeen of homogeen wees ?
4. Hoe moet die databasis versprei word betreffende die replisering en fragmentering van die databasis ?
5. Watter datamodelle, ontledingsbenaderings, metodieke ens. is van belang vir verspreide databasisontwerp ?

Hierdie vrae word gedeeltelik binne hierdie verhandeling beantwoord. Aspekte 1 en 2 word nie in hierdie verhandeling bespreek nie, maar aspekte 3, 4 en 5 word wel in 'n homogene fisiese databasis struktuur behandel (al die nodes binne die netwerk gebruik dieselfde tipe databasisbedryfstelsel (DBBS)). Hierdie beperking moet deurgaans in gedagte gehou word. So 'n databasis sal toegepas word op 'n organisasie met 'n verspreide struktuur.

Baie van die teorieë wat bespreek word, is gesetel in gesentraliseerde databasisontwerp, omdat daar huidiglik min inligting bestaan vir verspreide databasisontwerp. Daar sal gepoog word om die teoretiese gedeeltes wat in die lektuur op gesentraliseerde databasisontwerp van toepassing is, te projekteer na verspreide databasisontwerp. Dit is van toepassing op die meeste dele van hierdie verhandeling.

Die moderne mens funksioneer 8 ure per werksdag binne organisasies. Dit is dus belangrik om die omgewing waarin die moderne mens funksioneer, (die organisasie) te ontleed. Verder vorm inligting binne 'n organisasie een van die belangrikste en bruikbaarste hulpbronne, omdat die mens die meeste van die tyd daarmee werk. Die belang van verspreide inligting binne 'n organisasie kom ook

streng na vore, en word hierna bespreek.

1.2 DIE ROL VAN VERSPREIDE INLIGTING BINNE DIE ORGANISASIE

- Die rol wat verspreide inligting in die organisasie speel is afhanklik van die struktuur van die organisasie. James Martin [2] beklemtoon die volgende in sy boek "Information Engineering - Volume 2" :

"Information is a corporate resource and should be planned on a corporatewide basis regardless of the fact that it is used in multiple different computers and multiple departments"

Eerstens toon hierdie stelling die belangrike posisie wat inligting as hulpbron binne die organisasie inneem. Die sterkste klem is op inligting se rol as korporatiewe hulpbron. Hierom word die organisasie gedwing om vir inligting benutting te beplan, en dit daarna so effektief moontlik te bestuur. Sodoende vind ons regverdiging vir 'n databasis wat gebruik kan word om 'n organisatoriese inligtings-hulpbron te benut.

Verder noem James Martin dat hierdie inligting in terme van die hele organisasie benut moet word, ongeag van die geografiese of departementele verspreiding. As gevolg hiervan, moet die organisasie-eenheid in terme van fisiese verspreiding ontleed word. Prakties vind ons dat die organisasie-eenheid oor verskeie departemente en geografiese areas versprei is. Hierdie verspreidingsaspek regverdig dat baie aandag aan die implementering en ontwerp van die databasis geskenk moet word.

- 'n Verspreide databasis is die natuurlikste oplossing vir 'n probleem van verspreide inligtingsbenutting binne 'n organisasie. Ons sien dus dat die ontwerp van die databasis op konsepte wat deurgaans die verspreidingsveranderlikes in ag neem, gebaseer moet word.

- 'n Verdere analise van Martin se stelling toon dat inligting geïntegreer moet word ongeag die feit dat dit op verskillende rekenaars gestoor word. 'n Skakeling van hierdie rekenaars in 'n verspreide databasis sal verseker dat die verspreide inligting as 'n eenheid benut kan word. Sodoende kan ons algehele inligtingsbestuur toepas.

Teorey en Fry [33] ondersteun James Martin se siening van inligting as 'n korporatiewe hulpbron met die volgende stelling:

" The concept of the integrated database spanning multiple users was a direct result of the complex data structuring capabilities which the DBMS's afforded. The data can now be viewed as a corporate resource instead of as adjunct to a program, and consequently it should have an integrated (multi-user) requirement orientation instead of a single program orientation."

- Die doeltreffende gebruik van inligting as korporatiewe hulpbron, sal die organisasie se voortbestaan en groei verseker. Prakties vind ons dat effektiewe inligtingbenutting tot gevolg het, dat die organisasie makliker by veranderende omstandighede sal kan aanpas en meer kompetender sal kan funksioneer. As gevolg van die impak wat oneffektiewe inligtingbenutting op die groei en voortbestaan van die organisasie kan hê, moet die inligtings-hulpbron so effektief moontlik bestuur word.
- Die bestuur van inligting en die bestuur van organisatoriese databasisse kan as sinonieme begrippe beskou word, omdat groot gedeeltes van organisatoriese inligting vandag op rekenaars gestoor word. Daarom moet potensiële databasis strukture maksimaal ontwerp en benut word. Hierdie taak word gewoonlik deur die organisasie se inligtingsentrum (Information Centre) of rekenaarafdeling gehanteer.

In hierdie verhandeling gaan ons bestaande sentrale databasis ontwerpmetodieke, hulpmiddels en verspreide databasis konsepte kombineer in 'n raamwerk vir 'n metodologie vir verspreide databasisontwerp. Die verantwoordelikheid van die gebruik van hulpmiddels rus geheel en al op die organisasie se inligtingsentrum. Die sukses van enige databasis implementasie hang hoofsaaklik af van die noukeurigheid en integriteit van die ontwerpsprosesse wat deur die inligtingsentrum gevolg is.

- In verspreide databasisse is die ontwerpsfases uiters kritiek as gevolg van die verspreidings dimensie wat bykom. Hierdie verspreidings dimensie skep 'n bykomende veranderlike binne die ontwerpsproses. Die klem lê dus grootliks op hoe die ontwerp gedoen word, en watter benaderings, fases en metodieke die mees doeltreffendste sal wees inaggenome die verspreiding.
- Hierdie metodieke moet ook tyd-en-koste effektief wees omdat dit geregverdig moet wees binne die organisasiestruktuur. Die moontlikheid bestaan wel dat die organisasie se bestaande gesentraliseerde ontwerpstechnieke gebruik kan word om die verspreide databasis gedeeltelik te ontwerp.

1.3 GESENTRALISEERDE ONTWERP TEENOR VERSPREIDE ONTWERP

Ceri [1] [p68] noem dit dat in onlangse teoretiese navorsing in die verspreide databasis studiegebied, daar wel verspreidingskriteria en wiskundige fondasies vir die data-verspreidings probleem gevind is. Hy plaas die klem hoofsaaklik op die fragmentasie-en-verspreidingsaspekte van ontwerp.

Ceri noem ook dat enige sentrale databasisontwerpsmetodieke gebruik kan word vir die ontwerp van die verspreide databasis se globale skema. Dit wil voorkom dat hy hierdeur 'n groot gedeelte van die ontwerpsproses afskeep. Ons het tog te doen met die ontwerp van 'n verspreide databasis, en nie die

ontwerp van 'n gesentraliseerde databasis nie.

Die stelling, (dat verspreide en gesentraliseerde databasisontwerp grootliks dieselfde is) moet bevraagteken word deur dieper na die spesifieke elemente in verspreide databasisontwerp te kyk, en sodoende die mees gepaste verspreide databasis gebaseerde ontwerpmetodiek te vind.

Die "Infotech State of the Art Report, Distributed databases, Volume 1" [34] noem in een van die oop besprekings, (p 73) dat dit noodsaaklik is om 'n metodologie vir die ontwerp van - verspreide databasisse te skep. Die klem word dus op verspreide databasisontwerp geplaas, wat aandui dat dit inherent anders is as gesentraliseerde databasisontwerp. Hulle meen dat verspreide databasisontwerp baie moeiliker is as gesentraliseerde databasisontwerp. Daarom moet ons heelwat aandag skenk aan die verspreidings-aspekte van 'n ontwerpmetodiek vir verspreide databasisse omdat die verspreidings veranderlike (plek) glad nie binne gesentraliseerde databasisontwerp voorkom nie.

Om seker te maak dat die term 'metodiek' of 'metodologie' verder in die verhandeling duidelik verstaanbaar is, kan hierdie terme formeel omskryf word as: 'stapsgewyse prosesse wat verskillende benaderings implementeer om sodoende geïntegreerd, oorvleuelend, effektief en doelgerig te funksioneer'.

Die doeltreffende implementering van 'n databasis binne 'n organisasie, gaan hand aan hand met die suksesvolle benutting van inligting as hulpbron. 'n Suksesvolle ontwerpmetodiek kan gebruik word om die benutting van inligting as hulpbron binne die organisasie te maksimaliseer. Hiervoor kan 'n verspreide databasis uiters belangrik wees. Ons kyk vervolgens na die belangrike rol wat verspreide databasisse kan vervul, waarin regverdiging gevind kan word vir die implementering van 'n verspreide databasis.

1.4 - HOEKOM IS VERSPREIDE DATABASISSE BELANGRIK ?

'n Persoon wat in 'n organisasie funksioneer kan die vraag vra: " Het my organisasie 'n verspreide databasis nodig ?". Kleinrock [32] (p 92) noem dat 'n verspreide databasis geregverdig word waar die bron van die data versprei is en daar 'n behoefte vir lokale toegang tot die data is. 'n Organisasie met so 'n struktuur behoort dus 'n verspreide databasis te implementeer. In die algemeen moet die volgende aspekte altyd in gedagte gehou word wanneer implementering van enige databasis beplan word, nl:

1. Die struktuur van die organisasie.
 2. Kostevoordele.
 3. Topbestuur kan beheer maksimeer.
 4. Aanpasbaarheid tot veranderende omstandighede.
 5. Verhoogde sekuriteit.
 6. Makliker dataherwinning a.g.v. data oorbodigheid.
 7. Onafhanklike funksionering van stelsel-segmente.
- Ons gaan meer spesifiek na die volgende aspekte kyk omdat dit wel deel uitmaak van belangrike oorwegings by die implementering van 'n verspreide databasis nl:

1. Struktuur en geografiese verspreiding van die organisasie (punt 1 hierbo).
2. Koste voordele in terme van apparatuur en ander besparings (punt 2 hierbo).
3. Dinamiese aanpasbaarheid tot veranderende omstandighede (punt 4 hierbo).

Alhoewel bg. 3 aspekte baie belangrike kriteria is, moet daar in gedagte gehou word dat daar baie ander redes ook gevind kan word wat verspreide databasisse sal ondersteun. Net so is daar kriteria wat verspreide databasis implementering sal teewerk. In gevolg hierdie verhandeling gaan ons 'n paar

voordele van verspreide databasisse ontleed. Die basiese voordele wat verspreide databasisse inhou gaan hand aan hand met die volgende:

1.4.1 DIE STRUKTUUR EN GEOGRAFIESE VERSPREIDING VAN DIE ORGANISASIE

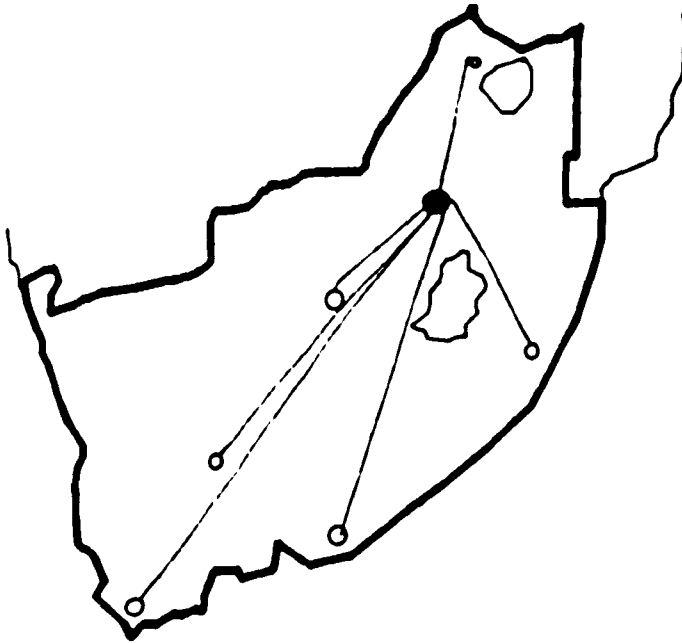
/ Tradisioneel moes die organisasie by die rekenaarstelsel aanpas a.g.v. die koste en tegnologiese implikasies wat betrokke was. Ons het gevind dat rekenaar-apparatuur relatief duur was en spesiale persele daarvoor aangelaë moes word. Hierdie rekenaar toerusting was ook meestal by die hoofkantoor gesetel waar net sekere van die toepassings gerekenariseer was.

Huidiglik is dit nie meer die geval nie. Die moderne benadering is om die rekenaarstelsel op die struktuur van die organisasie af te beeld. Dit impliseer verspreide databasisse as die organisasie verspreid is.

/ Tans vind ons dat die meeste organisasies groter desentralisasie ondervind. Ons vind ook dat korporatiewe oornames hierdie neiging verder aanwakker. 'n Verspreide databasis is dus ideaal om op die struktuur van 'n verspreide organisasie toe te pas. Verder kan die topbestuur met behulp van 'n verspreide databasis heeltyd beheer uitoefen oor die inligting in die organisasie. Dit word omvat in bestuursinligtingstelselteorie (Management Information Systems). Figuur 1 toon 'n vergelyking tussen 'n landswye verspreide databasis teenoor 'n landswye gesentraliseerde databasis wat op 'n verspreide organisasie afgebeeld is.

· Een van die eerste ontledingsaspekte, by die ontwerp en implementering van 'n verspreide databasis, is 'n analise van die fisiese verspreiding van die organisasie. Die

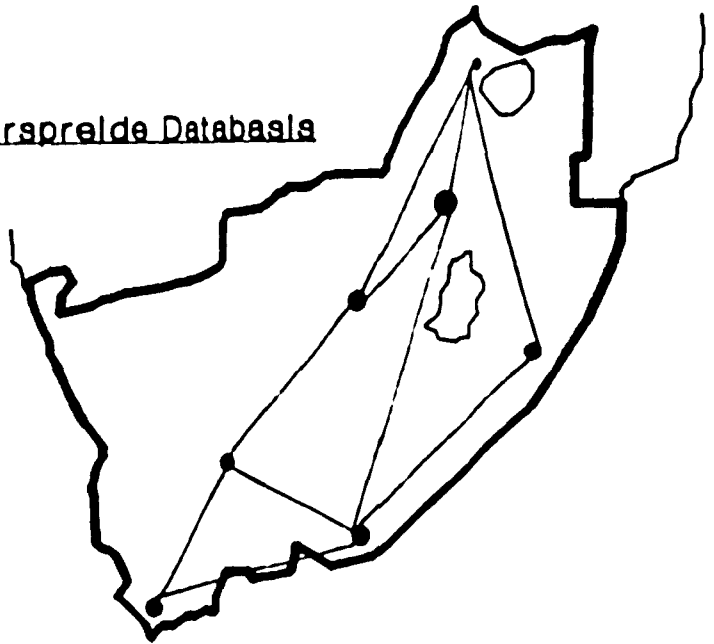
Gesentraliseerde Databasis



1. Databasis by sentrale node
2. Terminale by verspreide nodes
3. Een rekenaar by sentrale node
4. Kommunikasie vanaf sentrale na verspreide nodes.

Verspreide Databasis

1. Databasis by elke node
2. Terminale gekoppel aan lokale nodes
3. Kommunikasie tussen nodes gekoppel



Figuur 1 - Gesentraliseerde teenoor Verspreide databasis

ontwerpers moet hulself heeltemal van die fisiese verspreidingsaspekte, wat betrokke is, vergewis. Dit vorm later 'n baie belangrike invoer veranderlike by verskeie fases van die verspreide databasis ontwerpsproses.

- 'n Groot voordeel word aangetref waar 'n verspreide databasis op 'n verspreide organisasie afgebeeld word, omdat die struktuur van die organisasie nie by 'n gesentraliseerde stelsel hoef aan te pas nie. 'n Verspreide databasis kan ook makliker aanpas by dinamiese veranderinge wat in 'n verspreide organisasie plaasvind.

Tesame met die struktuur van die organisasie, moet ons ook na die aspekte kyk wat bestuursbesluite van die organisasie sal beïnvloed bv. koste aspekte betrokke by verspreiding en verspreide databasisse.

1.4.2 KOSTEBESPARINGS

Lampson, Paul en Siegert [30] meen dat tegnologiese vooruitgang na werklike kostebesparings lei. Hulle noem dat dit makliker en goedkoper is om 'n aantal verskillende rekenaars te koop, as om een komplekse multiprogrammeerbare verwerker met ekwivalente verwerkingskrag aan te skaf. Verspreide rekenaars kan met 'n kommunikasienetwerk verbind word. Kommunikasie-tegnologie is ook besig om met reuse spronge vooruit te gaan, bv. optiese verbindings, netwerke ens., wat beteken dat kommunikasieverbindings meer betroubaar word en die koste daarvan ook sal daal.

- Hulle noem ook dat verspreide verwerking ook kommunikasiekoste kan verminder omdat lokale take uitgevoer kan word waar dit lokaal benodig word. Verspreide databasisontwerp moet ook die outonome uitvoer

van take op lokale gedeeltes van die verspreide databasis en netwerkladings in ag neem. 'n Netwerk met ligte ladings het goedkoper kommunikasiekoste tot gevolg, in teenstelling met 'n swaargelaaide en duurder netwerk.

'n Ander kostevoordeel ontstaan uit die feit dat kleiner operasionele rekenaarsentrums tot beter geïmplementeerde projekte met kleiner projekspanne lei (verbeterde interprojek-kommunikasie).

Die kostebesparings wat 'n verspreide databasis bo 'n gesentraliseerde databasis mag meebring, behoort aanloklik vir enige organisasie te wees. Die regverdiging hiervoor stem uit die doel van die organisasie om die grootste wins met die minste moontlike uitgawes te maak.

Mendelson [37] (p12) noem in sy artikel "ECONOMIES OF SCALE IN COMPUTING: GROSCH'S LAW REVISITED" dat Grosch se wet nie meer van krag is nie. Grosch se wet noem dat die koste om X eenhede verwerkingskrag te bekom, proporsioneel afhanklik is van die vierkantswortel van X , d.w.s die koste per eenheid prestasie neem af teen 'n tempo van X tot die mag -0.5 . Deur Grosch se Wet as geldig te aanvaar, sal gesentraliseerde verwerking bo verspreide verwerking verkies moet word, a.g.v. die ekonomiese koste-implikasies van groot en kragtige rekenaars.

As Grosch se Wet huidiglik nog geldig was (Mendelson toon dat dit nie meer geldig is nie), sou ander redes vir verspreiding opgeweeg moet word teenoor koste besparings van sentralisasie, om sodoende verspreide databasisse te regverdig. Die praktiese implikasies van die teendeel van Grosch se Wet toon dat dit meer koste effektief sal wees om verspreide verwerkingskrag te gebruik. Mendelson toon dat sentralisasie van verwerkingskrag 'n verouderde beskouing van dataverwerking is, en nie noodwendig meer vandag van krag is nie.

Nadat Mendelson verskeie huidige artikels oor Grosch se Wet ontleed het, het hy tot die slotsom gekom dat in die 1980's 'n konstante gemiddelde koste per MIPS (Verwerkingskrag) bestaan. Hy het verskillende wiskundige en statistiese ontledings op huidige apparatuurkoste gedoen en tot 'n analitiese gevolgtrekking gekom. Sy slotsom is dat die koste faktor (Grosch koste) nou nie net meer geldig is om sentralisasie te regverdig nie, maar dat dit ook desentralisasie regverdig. Ons kan dus nie meer sê dat gesentraliseerde verwerking goedkoper as verspreide verwerking is nie.

Daarom moet ander voordele en nadele betrokke in die polimiek van sentralisasie teenoor verspreiding, ontleed word, om verspreiding te regverdig. Ander moontlike voordele dra meer gewig. So vind ons dat meer aandag aan die voordeel van aanpasbaarheid geskenk moet word om verspreiding te regverdig.

1.4.3 DIE AANPASBAARHEID VAN VERSPREIDE DATABASISSE TOT VERANDERING

Die veronderstelling word gemaak dat 'n verspreide stelsel, in teenstelling met 'n gesentraliseerde stelsel, baie meer aanpasbaar tot verandering sal wees. Die hoofrede hiervoor kan gevind word in die feit dat kleiner rekenaar-elemente makliker by lokale behoeftes aangepas kan word. In die gesentraliseerde geval vind ons dat die invloed van die verandering 'n nadelige uitkrangeffek op die stelsel het.

'n Verspreide databasis kan dus makliker by veranderende organisatoriese behoeftes aanpas, aangesien daar kleiner operasionele eenhede met outonome bestaansreg voorkom. In die praktyk kan elke databasisnode 'n eie dataverwerkingspan besit wat na spesifieke behoeftes kan

omsien mits die globale verspreide databasis in ag geneem word.

Yeh, Roussopoulos en Chang [47] noem dit dat 'n databasis ontwerpmetodologie behoort te verseker dat die databasis maklik aangepas kan word by die gebruikers se veranderende behoeftes. As gevolg hiervan behoort 'n databasis ontwerpmetodologie gebaseer te word op 'n verspreide databasis, omdat die struktuur van 'n verspreide databasis sal lei tot die grootste voordele in terme van aanpasbaarheid by veranderende omstandighede.

Vervolgens moet ons na die ontwikkeling van so 'n metodologie wat Yeh, Roussopoulos en Chang se vereistes van aanpasbaarheid sal kan hanteer, kyk. Die volgende afdeling hanteer die skep van 'n moontlike verspreide databasis ontwerpmetodologie, wat die voordele wat verspreide databasisse inhou, sal kan weergee.

1.5 DIE ONTWIKKELING VAN 'N VERSPREIDE DATABASIS ONTWERPSMETODOLOGIE

Ons gaan probeer om 'n ontwerpmetodologie vir verspreide databasisse te ontwikkel deur aspekte van bestaande ontwerpsteorië te ontleed. Die metodologie moet dus 'n aanpassing wees van die nuutste en bruikbaarste gedeeltes van die bestaande teorië betrokke by die verskillende aspekte van ontwerp. Ekstrapolasie van hierdie teorië moet gedoen word om die verspreidings-dimensie te omvat. Die aspekte betrokke by die ontwikkeling van 'n verspreide databasis ontwerpmetodologie is:

1. Die ontwerpsfases betrokke moet streng geïdentifiseer word. Deur noukeurig na fases te kyk, kan 'n objektiewe beeld van die ontwerpshandelinge wat gevolg moet word,

verkry word.

2. Die ontwerpstrategie wat gevolg moet word vir elke fase is belangrik. Hieronder vind ons die onder-na-bo, bo-na-onder en gekombineerde strategië.
3. Die modelleringsbenadering tot databasisontwerp moet gevolg word. Dit is een van die nuwe onderwerpe in databasisontwerp, en sluit die gebruik van die entiteitverwantskap en relasiemodelle in, om konsepuele en logiese modellering te doen.
4. Die spesifieke eienskappe wat 'n verspreide databasis anders maak as 'n gesentraliseerde databasis, moet altyd in die ontwerpproses ingebou word, bv. die verspreidings veranderlike (plek).

Deur die teorie in bg. aspekte te ontleed, aan te pas, en te gebruik, gaan ons poog om 'n fase georiënteerde metodologiese raamwerk vir verspreide databasisontwerp voor te stel. Eerstens kyk ons na die fases binne die tradisionele databasisontwerpsiklus, sodat dit aangepas kan word tot 'n verspreide databasisontwerpsiklus.

1.5.1 'n ONTLEDING VAN TRADISIONELE ONTWERPSFASES

Die ontwerp en implementering van 'n rekenaarstelsel word volgens Jensen en Tonies [46] (p36) in die volgende globale fases opgedeel:

1. Navorsingsfase
2. Konseptualiseringsfase
3. Ontwerp, ontwikkeling en implementering.
4. Toets-en-evalueringsfases
5. Operasionele-en-onderhoud fases

Jensen en Tonies doen deurgaans validasie en sertifisering van die ontwerp, behoeftes en die implementering. Hulle verseker dat die implementering wel die behoeftes bevredig. Dit behoort ook gedurende die ontwikkeling van 'n verspreide databasis gedoen te word. Wat interessant hieroor is, is dat die ontwerp van bo-na-onder geïmplementeer word terwyl validasie en sertifisering van onder-na-bo gedoen word. Deur die tradisionele programmatuur ontwikkelingsfases van Tonies en Jensen as uitgangspunt te gebruik, kan ons dit implementeer as basiese uitgangspunt vir verspreide databasisontwerp.

Ons gaan vervolgens 'n databasis ontwerpmetodiek in terme van bogenoemde fases definieer. Hierdie ontwerpmetodiek gaan net van toepassing wees op die eerste drie van Tonies en Jensen se fases, nl. die navorsingsfase, konseptualiseringsfase en die ontwerp-fase. Die eienskappe van die ontwerpmetodiek sal ook die onderhoud van die databasis vergemaklik asook die effektiwiteit van die onderhoudsfase (fase 5) verbeter, omdat databasisse nooit staties van aard is nie. Intendeel, ons vind dat databasisse as gevolg van nuwe behoeftes mettertyd verander. Die ontwerpbenadering of strategie wat gebruik word, moet hierdie dinamiese aspekte kan hanteer.

Die ontwerp-fases vir verspreide databasisontwerp wat Tonies en Jensen se fases met die modellerings benadering sal kombineer, sal later bekendgestel word. Ons maak dus seker dat die ontwerpbenadering wel moderne boublokke gebruik.

Voordat hierdie ontwerpbenadering vir verspreide databasisse voorgestel kan word, moet ons eers verskillende aspekte wat betrokke is by die ontwerp van 'n verspreide databasis, bestudeer.

Daar word algemeen aanvaar dat databasisontwerp hoofsaaklik uit die ontwerp van die globale konseptuele skema en meegaande subskemas bestaan (Oxborrow [38] (p95)). Ceri en Pelagatti [1] (p68) ondersteun dit en noem dat daar ekstra ontwerpsdimensies by verspreide databasisontwerp bykom, nl. die ontwerp van fragmentasie en die toewysing daarvan. 'n Groot gedeelte van die ekstra ontwerpskompleksiteit wat bykom, word gevind in die ontleding van die belangrikste toepassings sodat fragmentasieontwerp gedoen kan word. Ceri en Pelagatti wys daarop dat die globale skema en toepassingsontleding gedoen moet word in die proses van verspreide databasisontwerp.

In hierdie verhandeling gaan gepoog word om die raamwerk vir 'n ontwerpsmetodiek spesifiek vir verspreide databasisse te ontwikkel, sodat alle behoeftes van verspreide databasisontwerp op die mees effektiewe wyse opgelos kan word. Die klem sal op die ontleding en ontwerp van die databasis geplaas word ("uitputtende ontwerp"). As gevolg van verskeie tegniese verskille, word die ontwerp van 'n verspreide databasis as moeiliker beskou, vergeleken met gesentraliseerde databasisse. Daar moet onthou word dat die data en toepassings optimaal versprei moet word tussen die nodes van 'n netwerk om bevredigende databasisprestasie te verseker. Die klem verskuif dus na optimalisasie van die databasis.

Ontwerpaspekte betrokke by 'n verspreide databasis, waarin bogenoemde optimaliseringskwessie tevoorskyn tree, word deur die PACTEL korporasie, [42] (p27-35) genoem. Hierdie verspreide databasisontwerpaspekte is opgestel na 'n ondersoek van die praktiese en teoretiese aspekte betrokke by verspreide databasisontwerp, en behandel die volgende kwessies:

1. Herontwerp bestaande databasisse of begin nuut-van-voor-af. Die meeste organisasies besit reeds gesentraliseerde databasisse wat operasioneel is en konserwatief bestuur word. Dit gaan dus moeilik wees om 'n nuut-van-voor-af implementasie van 'n verspreide databasis te regverdig.
2. Segmentasie en duplisering van data. Waar moet data geplaas word en moet daar duplisering voorkom al dan nie? Verkeerde besluite kan die effektiwiteit en prestasie van die databasis beïnvloed.
3. Verdeling en plasing van beheer. Moet beheer gesentraliseer wees of ook versprei word? Gaan daar 'n sentrale beherende verwerker wees of verspreide outonome beheer?
4. Prestasie. Sal slegte ontwerp lei tot onaanvaarbare prestasie? Hoe ontwerp en implementeer ons vir optimum prestasie? Watter voordeel/nadeel-afspeling moet gedoen word om aanvaarbare prestasie te verseker?
5. Betroubaarheid. Sal die stelsel betroubaar wees en sal nodes betroubaar kan funksioneer? Waar is die swak skakels in die ketting?
6. Deursigtigheid vir gebruikers. Moet die gebruikers kennis dra van fisiese nodes / plekke waarmee hulle besig is, of moet dit deursigtig wees.
7. Databasis-struktuur en bedryfstelsel. Watter databasis moet gekies en gebruik word, en hoe gaan dit ineenskakel met ons kort- en langtermyn doelwitte?

8. Node kommunikasie. Watter kommunikasie metodes moet gebruik word vir inter-node kommunikasie bv. ISO/OSI vs. SNA met optiese verbindings ?
9. Graad van homogeniteit. Moet die databasis struktuur homogeen of heterogeen wees ? Kan ons 'n oorvleuelende model gebruik wat heterogeniteit deursigtig sal maak (bv. ANSI/X3/SPARC, IBM-SAA) in terme van apparatuur, databasisbedryfstelsels, logiese modelle, kommunikasieprotokolle ens. ?
10. Bou of koop 'n bestaande DBBS. Kan 'n kommersiële VDBBS (Verspreide DatabasisBedryfstelsel) die behoeftes bevredig of behoort 'n eie VDBBS ontwerp en geïmplementeer word ?

Ander aspekte wat bg. besluite moet kan komplimenteer, is organisatoriese aspekte (logiese en fisiese struktuur, administrasie, gebruikers, risiko-persepsie, sekuriteit, kennisvlak), ekonomiese aspekte (koste/prestasie, koste neigings van apparatuur, programmatuur, telekommunikasie, mense en geboue) en tegniese aspekte (kommunikasie-argitektuur, standardisasie en standaarde, data- vs. toepassingsanalise, telekommunikasiedienste en regerings wetgewing).

Binne 'n verspreide databasisontwerpsmetodologie vind ons na aanleiding van die vorige besprekings dat die volgende faktore aandag moet geniet:

1. Die verspreiding van die data.
2. Hulpmiddels vir die bepaling van verspreiding.
3. Ontwerp en analise hulpmiddels vir die skema/s.
4. Fragmentasie en toewysing van data.
5. Duplisering of replisering van data.

Vir die ontwerp van 'n verspreide databasis moet 'n raamwerk opgestel word wat ontwerpsdoelgerigtheid sal nastreef en

maksimaliseer. Die spesifieke fasette wat bykom vir verspreide databasisontwerp teenoor gesentraliseerde databasisontwerp, soos reeds genoem, is:

1. Ontwerp van fragmentasie (Globale relasies moet in terme van horisontale, vertikale en gemengde fragmente uitgedruk word.)
2. Toewysing van fragmente. (Replisering en fisiese toewysing.)

Ons onderskei hier tussen die logiese konseptuele aspekte van die databasis en die fisiese plasing van die data. Daar moet ook onthou word dat hierdie twee fasette ineengeskakel moet word, en nie onafhanklik van mekaar beskou moet word nie.

Alhoewel die ontleding van toepassings ook nuttig is vir gesentraliseerde databasisontwerp, is dit van kritieke waarde vir verspreide databasisontwerp. Net inligting i.v.m. die belangriker toepassings sal benodig word, omdat die inherente toepassingsonafhanklikheid van die databasis steeds geldig is. Die toepassingsgerigte inligting wat onder andere benodig word, is:

1. Oorsprong van uitvoering.
2. Frekwensie van uitvoering.
3. Hoeveelheid, tipe en verspreiding van toegang tot die data.

Deur bg. inligting in die ontwerpproses te gebruik, sal verskillende fragmentasie alternatiewe gevind kan word om sodoende optimale verspreiding te verseker. Twee verdere hoofstukke sal hierdie faktore grootliks behandel.

Die volgende ontwerpsmotiewe behoort deurgaans in gedagte gehou te word, (sommige is reeds genoem maar word weer deur Ceri en Pelagatti [1] {p69} beklemtoon):

1. Verwerkingslokaliteit: Maksimale lokaliteit, nl. plasing van data so na aan die gebruik daarvan as wat moontlik is. Ons vind dus lokale en verspreide verwysings na data. Hierdie motief kan vervul word deur lokale of verspreide verwysings te analiseer vir verskillende fragmentasie alternatiewes, en om die beste hiervan te kies.
2. Data beskikbaarheid en betroubaarheid: Vir "LEESNET" (Read Only) toepassings vind ons groter beskikbaarheid as daar 'n hoë mate van replisering voorkom. Betroubaarheid word onderhou deur duplikate by geografiese verspreide nodes te stoor. Replisering is dus uiters belangrik.
3. Verspreiding van die werklading: Verskillende verwerkers kan optimaal benut word. Parallele verwerking is ook nou moontlik. Ons vind hier 'n voordeel/nadeel-afspeling tussen werkladings-optimalisasie en verwerkingslokaliteit. 'n Maatstaf vir verwerkingspoed moet ook ingebou word.
4. Koste en beskikbaarheid van geheue: Elke node se geheue-optimalisasie moet in aanmerking geneem word betreffende koste- en beskikbaarheids-maatstawwe.

'n Ontwerpsmetodologie moet die belangrikste van bg. kriteria aanspreek. Vir verspreide databasisontwerp moet optimalisasie hoë prioriteit geniet.

Die bo-na-onder benadering word hoofsaaklik gevolg by die ontwikkeling van nuwe stelsels. Dit behels stapsgewyse verfyning van die globale skema tot by databasis-fragmentasie en toepassing implementering. Hierdie benadering word deur verskeie skrywers as baie rasioneel beskou, en is veilig om te volg.

Wanneer bestaande databasisse geïntegreer word, is die

onder-na-bo benadering van toepassing. Oorvleuelende data-definisies kan uitgeskakel word om een globale skema te vorm. Lokale data-definisies kan by die onder-na-bo en bo-na-onder gevalle geïmplementeer word om hulle plek in die globale skema in te staan. Die kwessie van onder-na-bo vs. bo-na-onder word verder in hierdie hoofstuk bespreek.

Gekoppelde databasisse met verskillende databasisbedryfstelsels (DBBS) skep probleme omdat daar een tot een vertalings tussen die datamodelle by die verskillende nodes, moet plaasvind. Hieroor moet 'n oorvleulende datamodel gevind word waarin die databasis ontwerp kan word, sodat hierdie model later omgeskakel kan word om die verskillende databasisargitekture te hanteer. Die omskakelings-koppelvlakke tussen die nodale datamodelle moet ook baie streng gedefinieer word. 'n Latere hoofstuk bespreek so 'n model (Entiteit-verwantskapmodel).

Onder-na-bo ontwerp (integrering van bestaande heterogene databasisse) benodig 'n oorvleulende konsepuele model (globale skema), 'n vertaler vanaf die lokale na konsepuele model en integrering van lokale modelle binne die globale skema, op fisiese sowel as logiese vlakke. Ons gaan hoofsaaklik die gebruik van die stapsgewyse bo-na-onder metodiek gebruik as illustrasie vir verspreide databasisontwerp omdat dit 'n getrouer beeld van die teoretiese konsepte betrokke by die ontwerp van verspreide databasisse sal lewer.

Die volgende gedeelte behandel die opstel van vereenvoudigde ontwerpfasies wat Tonies en Jensen se fasies sal integreer met die modellerings benadering. Die leser moet bg. ontwerpaspekte in gedagte hou, om sodoende nie die komplikasies betrokke by verspreide databasisontwerp te ignoreer nie.

Deur Jensen en Tonies [46] se tradisionele ontwerpfasas as raamwerk te neem, gaan ons 'n vereenvoudigde model vir die ontwerp van verspreide databasisse skep, waarby die modelleringsbenadering tot ontwerp gevoeg word. Hiervoor word die volgende fases geïdentifiseer:

1. Konseptuele Modelling (T & J Fases 1 en 2 nl. navorsing en konseptualisering)
2. Logiese Fisiese Modelling (T & J Fase 3 nl. ontwerp)
3. Implementering (T & J Fase 3 nl. implementering)

Die identifisering van bg. fases word ondersteun deur Hubbard [11] (p5) in sy boek "Computer Assisted Database Design". Hy identifiseer: konseptuele ontwerp, logiese ontwerp en fisiese ontwerp. Die verskil tussen die twee beskouings kom te voorskyn (in ons geval) in die feit dat die konseptuele modelleringsfase wel ook 'n mate van logiese ontwerp insluit, alhoewel ons nie fisiese ontwerp doen op die vlak waarop Hubbard dit aanbeveel nie (databasis algoritmes vir fisiese beheer van die data). Ons gebruik dus 3 basiese uitgangspunte vir elke fase nl. konseptueel, logies, en fisies, waarby 'n modelleringsbenadering bygevoeg is.

In die konseptuele modelleringsfase gaan ons eerstens 'n analise van die organisasie doen. Ons doen navorsing oor die werking van die organisasie. Hiervoor gaan ons hulpmiddels soos die organigram gebruik. Die totale organisasie moet ontleed word. Sodoende sal ons moontlike ontwerpse probleme vroegtydig diagnoseer. Hierdie analise berus op die feit dat goeie insig in die struktuur van die organisasie nodig is om die verdere ontwerpse fases suksesvol te kan voltooi. Die doel van hierdie fase is

om 'n konseptuele model van die struktuur en werking van die organisasie en die databasisbehoefte te skep.

In die logies-fisiese modelleringsfase gaan ons die konseptuele model op 'n fisiese implementeerbare databasismodel implementeer. Ons kan byvoorbeeld die konseptuele model transformeer na 'n relasiedatabasis.

Verskeie bronne nl. Hwang en Dayal [23], Setzer en Lapyda [24], en Dumpela en Arora [25] toon dat direkte vertaling tussen verskillende modelle wel moontlik is. Hulle noem spesifiek die moontlikheid van vertaling tussen die entiteitsverwantskapsmodel en die relasiemodel. Die doel van hierdie fase is om 'n databasismodel te skep wat die struktuur van die organisasie sal reflekteer.

Daar moet verstaan word dat die term "fisiese ontwerp" (soos deur Hubbard [11] genoem), verwys na 'n lae vlak databasisvoorstelling terwyl die term "logiese-fisiese model" verwys na 'n logiese model wat direk implementeerbaar is op 'n rekenaar, d.m.v. onder andere, die relasiemodel. Die koppeling met "fisies" ontstaan a.g.v. aanpassings by die logiese model wat gedoen moet word om 'n verspreide databasis te implementeer, nl. die verwysing na fisiese plekke.

In hierdie verhandeling verwys ons dus nie na die tradisionele beskouing van 'n fisiese model nie, (lae vlak lêer strukture) omdat ons aanneem dat dit deur die gekose databasisbedryfstelsel (DBBS) hanteer sal word.

Die fisiese gedeelte van die "logies-fisiese modelleringsfase" behels die implementeringsaspekte wat uniek aan verspreide databasisse is. Hierdie aspekte is die volgende: 1) Databasis fragmentasie en 2) Toewysing van fragmente. Die doel van hierdie fase is om die verspreiding-spesifieke aspekte van die implementering te hanteer en dit op ons logiese model toe te pas. Die

gebruik van die bo-na-onder benadering sal hierby ingesluit word. Daar moet onthou word dat hierdie fase op 'n logiese model geïmplementeer gaan word.

Die Implementeringsfase behels net die fisiese opstel van die databasis en die inwerkingstel van die verskillende gedeeltes van die databasis. Ons gaan nie veel aandag hieraan skenk nie, omdat dit nie meer deel uitmaak van die ontwerpaspekte van 'n databasis nie.

Teorey en Fry [33] het die stadiums in die lewensiklus van databasisse in terme van invoere en uitvoere geïdentifiseer (let op na die verwysing na toepassings). Die lyk soos volg:

INVOERE na die databasis ontwerpproses:

1. Onafhanklik van toepassings (Dek die langtermyn databasis behoeftes) nl: a) Gebruikers se beskrywings van die organisasie waarvoor data ingesamel moet word, b) Doelwitte van die databasis en c) Die gebruikers se siening van die data wat ingesamel en gestoor moet word.
2. Afhanklik van toepassings nl: a) Die spesifieke data-elemente wat by die toepassings benodig word, b) Data volumes en c) Verwerkingsfrekwensies van toepassings (uitvoering per tydeenheid).

UITVOERE van die databasis ontwerpproses:

1. Logiese databasis struktuur.
2. Fisiese databasis struktuur.
3. Spesifikasies vir toepassings wat op die databasis strukture uitgevoer gaan word om behoeftes te bevredig.

Die feit dat hulle toepassings as deel van die ontwerp gespesifiseer het, beteken dat toepassings 'n pertinente rol binne databasisontwerp speel, en dus in 'n ontwerpsmetodiek behandel moet word. Die metodologiese raamwerk wat later in hierdie hoofstuk voorgestel word, neem ook toepassings in ag.

Die benaderings wat reeds genoem is, pas in by die basiese lewens-siklus vir databasis implementering soos ook deur Teorey en Fry genoem is in "Infotech State of the Art Report: DISTRIBUTED DATABASES Volume 1: Analysis & Bibliography (p80)". Dit is:

1. Behoeft-formulering en analise.
2. Logiese ontwerp.
3. Fisiese ontwerp en evaluasie.
4. Implementering.
5. Operasionele toepassing en monitor.
6. Veranderinge en aanpassings.

'n Modelleringsbenadering pas dus korrek in by lewens-siklus omdat die terme "konseptueel" gedek kan word in "behoefte formulering en analise" en die terme "logies" en "fisies" ook voorkom. 'n Ontwerpsmetodologie wat Tonies en Jensen se tradisionele fases, die modelleringsbenadering en Teorey en Fry se siklusse sal ondersteun, kan ge-sintetiseer word na die volgende:

1. Ontleding, analise en formulering van die behoefte.
2. Konseptuele modellering.
3. Toepassingsanalise en ontwerp.
4. Logies-Fisiese modellering.

Ons sal hierdie fases later kombineer met spesifieke elemente wat betrokke is by die ontwerp van verspreide databasisse. Daarna sal ons die benadering wat gevolg word, inbou in elke stap, om sodoende 'n volledige metodologiese raamwerk daar te stel. Om die gebruik van

datamodelle in die ontwerpsproses te regverdig, moet ons, onself van die rol wat datamodelle kan speel, vergewis.

1.5.4 DIE ROL VAN DATAMODELLE IN DIE ONTWERPSPROSES

Die belangrikste redes vir die gebruik van datamodelle in die ontwerpsproses is die volgende (Soos deur verskillende skrywers voorgestel):

1. Verbeter kommunikasie tussen die gebruiker, ontwerper en implementeerder. Modelle is maklik verstaanbaar en verskaf 'n abstrakte, maar verstaanbare beeld van die realistiese werklikheid.
2. Dit dwing die ontwerper om noukeurige ontwerp toe te pas. Daar is spesifieke stappe wat gevolg word in die opstel van die model en net 'n eindige aantal boublokke kan gebruik word.
3. Dit dien as 'n basis vir verdere ontwikkeling en uitbreiding van die databasis.
4. Die konseptuele model kan vir ander doeleindes gebruik word bv. topbestuur se herstruktuering van die organisasie.
5. Die gebruik van modelle kan moontlike probleme in die werking van die organisasie uitwys, bv. bottelnekke in 'n produksiekanaal.
6. 'n Oorvleuelende model kan gebruik word om verskillende modelle te beskryf bv. 'n entiteitverwantskapmodel kan 'n heterogene verspreide databasis wat bestaan uit relasie-, netwerk- en hierargiese-modelle beskryf.

Die gebruik van modelle bied dus groter doelgerigtheid aan behoeftebevrediging. Dit bied ook 'n tussenvlak vir die uitruiling van idees. Verder kan 'n komplekse ontwerp ook makliker verstaan word deur 'n model daarop toe te pas, om sodoende die kompleksiteit te verminder.

'n Datamodel kan gedefinieer word as: 'n diagrammatiese en/of semantiese voorstellingshulpmiddel wat die realistiese werklikheid modelleer. Tsichritzis en Lochovsky [9] {p5} beskryf 'n datamodel as 'n intellektuele abstraksiemiddel wat eenvoudig genoeg is om kleinighede i.v.m. die realistiese werklikheid aan te toon, en kragtig genoeg is om verwantskappe tussen die data uit te wys. Die volgende van hulle stellings beskryf datamodelle in eenvoudige terme:

" A data model is an abstraction device that allows us to see the forest (information content of the data) as opposed to the trees (individual values of data)."

1.5.4.1 DATAMODEL KLASSE EN VOORSTELLINGS

Korth en Silberschatz [39] {p6} definieer 'n datamodel as 'n versameling konseptuele hulpmiddels wat gebruik kan word om data, data-verwantskappe, data-semantiek en databeperkings te beskryf. Met betrekking tot databasisontwerp, verdeel hulle datamodelle in twee klasse nl :

1. Objek-gebaseerde logiese modelle.
2. Rekord-gebaseerde logiese modelle.

In terme van ontwerp kan die rol wat datamodelle in die ontwerpsproses speel, beskryf word as hulpmiddels wat kan lei tot maksimale bevrediging

van die ontwerpbehoefte. Ons gaan in hierdie verhandeling 'n objek-gebaseerde model en 'n rekord-gebaseerde model binne 'n ontwerpmetodologie voorstel, en toon dat so 'n kombinasie uiters geskik is vir die ontwerp van verspreide databasisse.

Die modelle wat gebruik gaan word, is onderskeidelik die objek-gebaseerde entiteitverwantskapmodel en die rekord-gebaseerde relasiemodel. Hierdie twee modelle gaan later in verdere besonderheid bespreek word. 'n Vinnige samevatting van die ander modelle wat in elkeen van die twee kategorië beskikbaar is, sal later getoon word. Let op dat albei modelle logiese modelle is.

Definisies moet eers verskaf word vir die terme "objek-gebaseerde logiese model" en "rekord-gebaseerde logiese model" (Korth en Silberschatz [39] [p6,7]). 'n Objek-gebaseerde model is 'n datamodel wat gebruik kan word om die logiese konseptuele behoeftes van die gebruiker met die begrip van die ontwerper te konsolideer deur werklike objekte (mense, motors, prosesse) te gebruik. So 'n model is konseptueel beskrywend. Die data-objekte (entiteite) en verwantskappe daartussen, word hoofsaaklik gebruik om voorstellings te skep. Die model toon ook data kardinaliteite tussen objekte (1:1 of 1:N of N:N).

'n Rekord-gebaseerde model kan gedefinieer word as 'n datamodel wat 'n databasis in terme van die konseptuele struktuur van die data uitbeeld. Die model is dus 'n gestruktureerde model wat fisies op 'n rekenaar implementeerbaar is en werklike verwysings na data bevat, soos bv. 'n naam van 'n persoon. Hierdie tipe model vertoon gewoonlik nie eksplisiete kardinaliteite nie, maar dit word d.m.v.

werklike voorkomste van data geïmplementeer.

Ons kan verder onderskeid tref tussen 'n werklike fisiese model (laevlak dataleërs, indekserings tabelle, toegang en sortering algoritmes) en die rekord-gebaseerde logiese model, wat ook fisiese data bevat. Die rekord-gebaseerde model word gebruik om die rekenaarimplementering van 'n objek-gebaseerde model te doen, bv. die entiteit-verwantskapmodel kan fisies as 'n relasiemodel implementeer word. Ons sal later sien hoe dat hierdie twee modelle sekvensieël ineenskakel d.m.v. transformasiealgoritmes.

Ons gebruik dus 'n objek-gebaseerde model as die ontwerpmodel vir die implementering van 'n rekord-gebaseerde model. Die rekord-gebaseerde modelle gebruik fisiese modelle om die apparatuur van die rekenaar te benut. Die laevlak fisiese model is nie van toepassing in hierdie verhandeling nie, omdat dit binne databasisbedryfstelselteorie behandel word. Ons het tot dusver gekyk na fases vir databasisontwerp en die gebruik van modelle. Nou gaan ons vlugtig na ontwerpstrategie kyk.

1.5.5 ONTWERPSTRATEGIE VIR VERSPREIDE DATABASISSE

Vervolgens kyk ons na die rigting waaruit die ontwerper die probleem moet benader, oftewel die ontwerpstrategie wat gevolg moet word. Hierdie strategie gaan 'n inherente deel van die finale ontwerpmetodologie vorm. In besonder, wat van belang is, is die bo-na-onder en gekombineerde benaderings.

Teorey, Yang en Fry [31] meld dat tradisionele databasisontwerpsbenaderings hoofsaaklik onder-na-bo

benader was, waarby data-elemente gekombineer was in genormaliseerde relasies, wat funksionele afhanklikhede tussen data reflekteer. Hulle meld ook dat die kompleksiteit en grootte van die databasisontwerpsproses die praktiese implementering van 'n ontwerp kan kortwiek. Sodoende word volledige ontwerp nooit effektief toegepas nie.

Verder word gemeld dat die gekombineerde benadering soveel meer aanvaarbaar is a.g.v. die populariteit van die entiteitverwantskapmodel. Die onder-na-bo en bo-na-onder strategië vir verspreide databasisontwerp sal nou in meer besonderheid behandel word.

1.5.5.1 BO-NA-ONDER ONTWERP

Freeman [7] is van mening dat hierdie benadering oor die algemeen meer gebruik word vir die ontwerp van nuwe stelsels. Hy definieer die bo-na-onder benadering as: 'n groepering van besluite in verskillende vlakke, waarvan die hoogste-vlak-besluite eerste geneem word. Hy noem verder dat die besluite binne elke vlak, die laerliggende alternatiewe so min as moontlik moet beïnvloed. In die praktyk vind ons dat die beplanning en ontwerp van 'n nuwe verspreide databasis volgens die bo-na-onder metode gedoen moet word, omdat dit stapsgewys verfyn kan word en meer besonderhede met elke stap bygevoeg word. Hierdie punt word ondersteun deur Adiba [14] wat dit 'n T-stelsel (Top-Down) noem.

1.5.5.2 ONDER-NA-BO ONTWERP

Die onder-na-bo benadering is die ideale benadering om bestaande databasisse mee te integreer. Ons benader die integrasie deur om eerstens die laagste vlak elemente met mekaar te kombineer en oorvleueling uit te skakel. Die grootste nadeel van hierdie benadering is dat dit nie gebruik kan word om 'n databasis vanaf 'n konseptuele model te implementeer nie.

Denning [8] beskryf die onder-na-bo benadering as: 'n stapsgewyse redusering van elemente wat van die laagste vlak na die hoogste vlakke toe beweeg. Dit is ideaal om losstaande gesentraliseerde databasisse te kombineer tot 'n verspreide databasis, omdat dataoorbodigheid en oorvleueling vanaf onder-na-bo uitgeskakel en gekombineer kan word.

Adiba [14] noem dat verspreide databasisse wat ontstaan deur databasisse by verspreide nodes te bou en dit later te integreer, 'n B-stelsel (Bottom-Up) genoem word. Dit beteken dat nodegebonde databasisse, geïntegreer word na een verspreide databasis van onder-na-bo.

1.5.5.3 KOMBINASIE VAN BO-NA-ONDER EN ONDER-NA-BO BENADERINGS

Ons kan toon dat die onder-na-bo benadering ideaal is om bestaande databasisse te integreer, en die bo-na-onder benadering gebruik kan word om nuwe databasisse te ontwerp. 'n Kombinasie van die twee kan ook in 'n ontwerpsmetodologie geregverdig word. Teorey, Yang en Fry [31] noem pertinent dat die bestaansreg van die gekombineerde benadering gesetel

is in die aanvaarding en gebruik van die entiteitverwantskapmodel. Dit bied aan die ontwerper 'n bruikbare konseptualiseringsmiddel. Die gebruik van beide die bo-na-onder en onder-na-bo benaderings sal volledige en geïntegreerde ontwerpstrategie tot gevolg hê, mits dit reg gebruik word. Prakties, in verspreide databasisontwerp, kan die bo-na-onder benadering gebruik word om die toepassings en objek gebaseerde gedeeltes te ontwerp. Die onder-na-bo benadering kan dan gebruik word om die datastrukture te optimaliseer en te fragmenteer.

Ons het reeds aangevoer dat die gebruik van modelle heelwat voordele vir die ontwerper inhou. Ons het ook ontwerpstrategie en fases beskou. Die vraag ontstaan nou:

"Watter datamodelle moet gekies word om verspreide databasis ontwerpfasies te ondersteun?"

Aanbevelings wat lei tot die beantwoording van hierdie vraag, word in die volgende afdeling behandel.

1.5.6 DIE KEUSE VAN DATAMODELLE VIR VERSPREIDE DATABASISSE

Omdat ons die klem so sterk op die gebruik van modelle in die ontwerpproses geplaas het (Afdeling 1.5.4 - Die rol van datamodelle in die ontwerpproses - p26), is die keuse van modelle uiters belangrik. Ons moet dus model/le kies wat in toepaslike verspreide databasis ontwerpfasies van toepassing gaan wees. As ons suiwer na datamodel teorie kyk, vind ons dat Tschritzis en Lochovsky [9] 'n magdom datamodelle identifiseer nl:

1. Relasiemodel
2. Netwerkmodel
3. Hiërargiese model
4. Entiteitverwantskapmodel
5. Binêre model
6. Semantiese netwerkmodel
7. Infologiese model

Hulle beskou 'n datamodel as : 'n voorstelling wat data strukturerings reëls definieer. Die toepassing van 'n datamodel op 'n situasie (probleem) vorm 'n skema. Hierdie skema kan byvoorbeeld objekte soos persone bevat wat dan beskryf kan word in terme van naam, tipe, ouderdom, gewig, ens. Dit sluit dus aan met ons vorige bespreking van datamodelle. As gevolg van die hoeveelheid modelle en permutasies van elkeen wat bestaan, is die keuse van modelle nie maklik nie.

Die twee modelle wat ons gaan gebruik vir konseptuele-logiese modellering en logies-fisiese modellering, is onderskeidelik Chen [22] se entiteitverwantskapmodel en Codd [3] se relasiemodel. Hierdie keuse beteken nie dat die ander modelle nie 'n rol in verspreide databasisontwerp kan speel nie, intendeel, die ander modelle kan net so effektief wees. Ons moet net altyd daarvan seker wees dat die gekose model reg toegepas word, en nie gebruik word buite sy definisie nie.

Die keuse van die entiteitverwantskapmodel en relasiemodel berus grootliks op die gewildheid van beide modelle. Ons gaan nou kyk na redes wat die keuse van hierdie modelle vir verspreide databasisontwerp, sal beïnvloed.

Vir konseptuele-logiese modellering, word die entiteitverwantskapmodel gekies, a.g.v. die volgende redes:

1. Chen [22] noem dat hierdie model eenvoudig is en dat dit kommunikasie tussen die ontwerper en gebruiker vergemaklik.
2. Die model is kompleks genoeg om alle realistiese en werklike situasies voor te stel.
3. Die struktuur van die model is aanpasbaar binne wyd gespesifiseerde grense en reëls.
4. Die model gebruik hoofsaaklik die konsep van stapsgewyse verfyning (bo-na-onder ontwerp) wat dit baie effektief maak, omdat dit die hoeveelheid entiteite onder beskouing vermeerder terwyl die kompleksiteit van die model toeneem.
5. Die model kan generiese beskrywings van ander modelle en beskrywings van die realistiese werklikheid hanteer.

Vir logies-fisiese modellering ('n implementeerbare model) is die relasiemodel gekies omdat dit oor die volgende eienskappe beskik:

1. Eenvoudige datavoorstelling.
2. Kragtige databasis manipuleerstrukture.
3. Wye navorsingsbasis, veral in normalisasieteorie en optimalisasie van databasisse.
4. Teorey, Yang en Fry [31] meld dat die struktuur en formaliteit van die relasielatedatabasis ideaal is, spesifiek in terme van 'n teikenmodel vir transformasie vanaf die entiteitverwantskapmodel.

'n Verdere rede wat aangevoer word vir die keuse van hierdie twee modelle, is die feit dat hulle fisies maklik

met mekaar kan integreer. Daar sal ook ander redes vir die gebruik van die twee modelle te voorskyn kom in latere bespreking.

Deur die entiteitverwantskapmodel as aanknopingspunt vir databasisontwerp te gebruik, voeg ons 'n ekstra ontledingsdimensie by die ontwerpsproses. Dit beteken ook dat ons makliker die ontwerp van heterogene verspreide databasisse kan hanteer, omdat die entiteitverwantskapmodel gebruik kan word om na ander fisies implementeerbare modelle (netwerk, hiërargies en relasie) te transformeer.

Ons gebruik die entiteitverwantskapmodel as ontledingsmodel, terwyl ons die relasiemodel as implementeringsmodel gebruik. Deur, byvoorbeeld die relasiemodel as die enigste model te kies, sou ons sekere konseptualiserings behoeftes van die ontledingsproses nie volledig kon bevredig nie. Daar gaan spesifiek 'n hoofstuk aan elk van die twee modelle bestee word, waarin die toepassing van elk op verspreide databasisontwerp, behandel sal word.

Ons het reeds besluit watter modelle ons gaan gebruik, nl. die entiteitverwantskapmodel en die relasiemodel. In die volgende twee afdelings, gaan ons kyk na die basiese elemente waaruit hierdie modelle bestaan en hoe hulle lyk, sodat die leser vroegtydig beide modelle kan visualiseer voordat die volgende hoofstukke behandel word. Die entiteitverwantskapmodel word eerstens behandel.

1.5.6.1 DIE ENTITEITVERWANTSKAPMODEL

Hierdie model se oorsprong is te danke aan P.P. Chen (22) wat die EV model in sy artikel: "The entity-relationship model: Toward a unified view of data", beskryf. Die motivering was om 'n model te skep wat as 'n standaard vir skema-definisie gebruik kan word. Vanaf 1976 het verskeie permutasies van hierdie model ontstaan, wat later in 'n raamwerk getoon sal word (hoofstuk 4).

Ons vind ook bv. dat James Martin (2) EV-modelle (EV = entiteitsverwantskap) in sy praktyk-georiënteerde boek: "Information Engineering - Vol 2", gebruik. Hy word wêreldwyd erken as: "James Martin - 'n meester in die praktiese toepassing van teorieë." Die EV-model is dus ook belangrik omdat James Martin sy naam daaraan koppel (Sommige in die rekenaarindustrie sal nie hiermee saamstem nie maar die idee bly steeds geldig). Ons kan dus die stelling maak dat sy keuse van die entiteit-verwantskapmodel nie geïgnoreer kan word nie.

Ons gaan later kyk na 'n uitgebreide entiteits-verwantskapsmodel wat met groot vrug op verspreide databasisse toegepas kan word (hoofstuk 4). Hierdie model gaan gebruik word om 'n konseptuele skema te ontwikkel wat die spesifieke behoeftes van verspreide databasisontwerp sal kan bevredig.

Die basiese boublokke van die Entiteits-verwantskapsmodel bestaan uit die volgende:

1. Entiteite en entiteit-tipes: Entiteite beskryf objekte of elemente binne die realistiese werklikheid waarvoor inligting ingesamel kan word. 'n Versameling soortgelyke entiteite vorm 'n entiteit-tipe. Entiteit-tipes beskryf 'n generiese

groepering vir entiteite bv. entiteit-tipe "MASJIEN" beskryf entiteite "IBM REKENAAR", "MERC DIESEL ENJIN", "CASE SKOPGRAAF". 'n Entiteit kan in meer as een entiteit-tipe voorkom, bv. 'n entiteit "DOKTER" kan ook in die entiteit-tipe "PASIËNT" aangetref word. Soms verdwyn die definisieverskil tussen entiteit-tipes en entiteite sodat net die term "entiteit" as sinoniem gebruik word.

2. **Verwantskappe:** Dit beskryf die verhouding waarin die entiteite teenoor mekaar funksioneer. Verwantskappe bestaan uit verwantskaptipes wat beskrywende inligting, i.v.m. die verbindings tussen entiteite bevat. Primêr beskryf dit die verhouding tussen entiteite bv. die entiteit "DOKTER" kan 'n 0:N (nul tot meer) "ONDERSOEK" verwantskap teenoor die entiteit "PASIËNT" openbaar. Terugwerkend kan entiteit "PASIËNT" 'n 1:M (een tot meer) "WORD-ONDERSOEK-DEUR" verhouding teenoor entiteit "DOKTER" openbaar. Dit beteken dat 'n dokter geen pasiënte of meer as een pasiënt kan hê terwyl 'n pasiënt deur een of meer dokters besoek kan word.

3. **Entiteitverwantskapsdiagramme:** Dit verskaf 'n diagrammatiese voorstelling van entiteite en hulle verwantskappe. 'n Grafiese voorstelling kan baie makliker geïnterpreteer word, omdat dit 'n duideliker geheelbeeld kan vorm. 'n Diagrammatiese voorstelling is die belangrikste eienskap van die EV-model. 'n Voorbeeld word getoon in figuur 2.

4. **Waardestelle en attribute:** 'n Waardestel bestaan uit die basiese elemente of eienskappe waaruit 'n verwantskap of 'n entiteit kan bestaan, bv. 'n attribuut "ADRES" kan entiteit "WERKNEMER" met waardestelle "STRAAT-NAAM" en "WOONAREA" beskryf. 'n Waardestel bestaan uit verskillende waardes bv. 'n waardestel "VLEIGTUIG-NOMMERS" kan uit waardes

112, 113, 114, 115, 116 bestaan. Beperkings op waardestelle moet aangetoon word bv. "HOTELKAMERS" kan net uit nommers 1 tot 12 en 14 tot 32 bestaan.

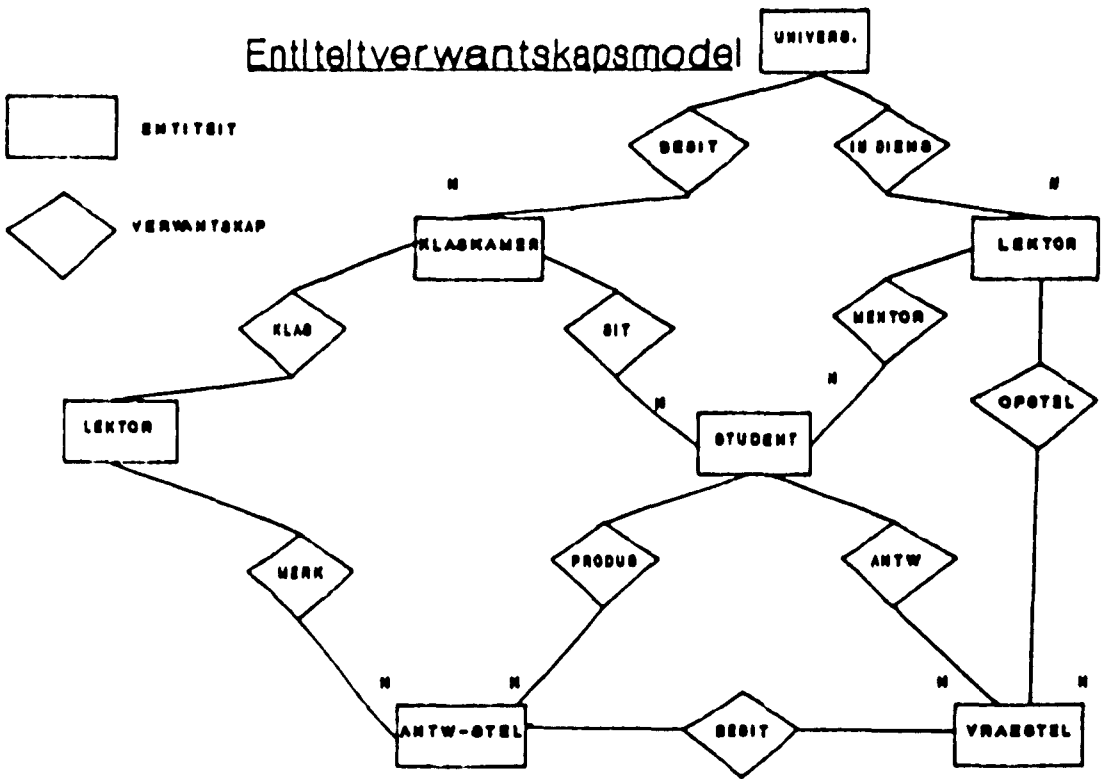
5. Swak entiteite: 'n Swak entiteit beskik oor dieselfde definisie as 'n normale entiteit, behalwe dat 'n swak entiteit net kan voorkom as daar reeds 'n eenaar entiteit bestaan, bv. 'n swak entiteit "ONDERSOEK REKORD" is onderhewig aan die bestaan van entiteit "PASIENT". Die verwantskap wat dan die normale entiteit met die swak entiteit verbind, word ook 'n swak verwantskap genoem.

'n Beskrywende voorbeeld van 'n tipiese EV diagram kan in figuur 2 gevind word. Hierdie EV diagram beskryf die uitleg van 'n universiteit. 'n Volledige beskrywing van 'n uitgebreide en verfynde EV-model vir verspreide databasisse sal later in hierdie verhandeling bespreek word (hoofstuk 4). Die ander belangrike datamodel wat gekies is vir hierdie verhandeling, is die relasiemodel.

1.5.6.2 DIE RELASIEMODEL

Die ontstaan van die relasiemodel is te danke aan E.F. Codd [3] met sy geskiedkundige werk : " A Relational model for Large Shared data banks". Hierdie kontroversiële werk het baie van die destydse (1970's) benaderings omver gewerp. Dit het tot gevolg gehad dat die databasis kennisveld in 'n nuwe rigting beweeg het. Die inpak daarvan word grootliks gesien op die logiese beskrywing en manipulasie van data in terme van eenvoudige plat lêers (Cardenas [4] (p99)).

Entitelverwantskapsmodel



Relaslemodel

STUDENT

NOMMER	NAAM	KURSUS
78/8888/7	Mar Andewich	Beema
88/8878/8	Mar Berthelma	Beema
48/8764/1	Mar Olofenda	BA
48/8812/8	Mar Gumbel	Beo
88/8718/8	Mar Jellene	BA
88/8877/4	Mar Pistoro	Beo

LEKTOR

ID#	NAAM	VAK
1000	Mar Andromeda	Biologie
2000	Mar Sotocell	Wetreg
4000	Mar Sellard	Dierkunde
8004	Mar Vessendel	Rechtsges

Figuur 2 - Entitelverwantskap- en relaslemodel diagramme

Huidiglik (1988) is daar baie implementasies van die relasiemodel bv. IBM se SQL/DS wat in 1981 reeds bekend gestel is, IBM DB2, en die ORACLE DBBS. Die relasiemodel beskou die logiese struktuur van die databasis as 'n versameling van twee-dimensionele tabelle of relasies.

'n Kolom in hierdie tabel word 'n domein genoem. Een ry van die relasietablel word 'n tal genoem. 'n Enkele datawaarde wat in 'n domein voorkom, word 'n veldwaarde genoem. Elke tabel beskik oor 'n unieke identifikasie, nl. 'n relasiename. Die belangrikste eienskap van die relasies is dat geen herhalende datatalle daarin mag voorkom nie. Elke domein het ook 'n unieke naam.

Die struktuur van 'n databasis word in terme van bg. boublokke uitgebeeld. 'n Diagrammatiese voorstelling kan in figuur 2 gevind word.

Codd [3] het verder twee verskillende relasi-datamanipuleertale voorgestel. Hy noem dit relasi-algebra en relasierekening. 'n Goeie implementering van relasi-algebra kan in IBM se SQL/DS gevind word.

Die keuse van die relasiemodel om 'n verspreide databasis mee voor te stel, is geregverdig deur die feit dat dit wel huidiglik as 'n fisiese-logiese model gebruik kan word, en dat dataverwysings onafhanklik van fisiese strukture is. Die belangrikste bewerking wat op die ontwerp van relasiemodelle gedoen word, is normalisasie. Normalisasie kan beskryf word as: 'n metode wat gebruik kan word om herhalende groepe data te verwyder en funksionele data-afhanklikhede uit te wys. Datamanipulasie kan d.m.v. relasi-algebra gedoen word.

Ons vind dat daar vyf relasie-algebra operatore bestaan nl: 1) Samevoeging, 2) Projeksie, 3) Samestelling, 4) Beperking en 5) Deling.

Net soos die veranderlikes in 'n gewone programmeertaal gebruik kan word in 'n toewysing soos $A = B + C$, word relasies d.m.v. relasie-algebra gemanipuleer. So kan ons 'n nuwe relasie "DEPT-PERS" skep deur relasies "DEPT" en "PERS" saam te voeg, oor domein "PERSNOMMER". Dit kan voorgestel word deur die stelling DEPT-PERS = PERSNOMMER [PERS * DEPT]. Hierdie voorbeeld illustreer die gebruik van die samevoeging (*) operator.

Die projeksie operator (II) doen net die teenoorgestelde van die samevoeging operator. Dit reduceer relasies na 'n nuwe relasie wat net gespesifiseerde domeine bevat. Dit is veral nuttig waar 'n persoon nie toegang het tot inligting in al die domeine binne 'n relasie nie. 'n Voorbeeld hiervan is, as 'n relasie "SOLDAAT-ID" met domeine ("MAGSNO", "NAAM", "VAN") uit die oorspronklike relasie "SOLDAAT" met domeine ("NO", "NAAM", "VAN", "KODENAAM", "REGIMENT") gevorm word. Die relasie algebra word voorgestel as:

SOLDAAT-ID = II SOLDAAT(NO,NAAM,VAN).

Die samestelling operator gebruik in werklikheid die projeksie en samevoeging operators. Twee relasies word saamgevoeg oor 'n sekere domein, daarna word dit geprojekteer.

Die beperking operator word gebruik om een relasie uit twee verskillende relasies te vorm. Net die rye, waarvan waardes in die gespesifiseerde domeine ooreenstem, word in die nuwe relasie geplaas. 'n Voorbeeld hiervan is, waar 'n relasie "MASJIEN-NR" beperk word teenoor relasie "OPERATEUR" met

oorvleulende domein "MASJIEN-ID", om 'n nuwe relasie te vorm, wat inligting sal bevat oor die masjiene wat aan operateurs toegeken is.

Die deling operator (/) word gebruik om 'n relasie te vorm deur twee ander relasies met mekaar te deel. As ons kyk na die stelling PERSOONLIKE-DOKTER = DOKTER / PASIËNT waar die relasie "DOKTER", twee domeine bevat, ("DOKTER-ID", "PASIËNT-ID") en die relasie "PASIËNT" een domein met waardes bevat, naamlik ("PASIËNT-ID"), bestaan die resulterende relasie "PERSOONLIKE-DOKTER" uit domein ("DOKTER-ID"), wat net die ooreenstemmende waardes tussen die twee relasies se ("PASIËNT-ID") domeine bevat. 'n Voorbeeld van so 'n vraag is bv. "Verskaf die name van die spesialiste wat pasiënt P113324 ondersoek het".

Relasie-algebra operators plaas uiters kragtige manipulasiestrukture in die hande van programmeerders en gebruikers. Die eienskappe van die relasiemodel word d.m.v. die struktuur, normalisasie en relasie-algebra uitgedruk. Ons gaan later hierdie aspekte in terme van verspreide databasisse beklemtoon (hoofstuk 6).

Nou dat ons reeds die gekose datamodelle bespreek het, en ook gekyk het na ontwerpsfases en ontwerpsbenaderings, word daar in die volgende gedeelte 'n poging aangewend om 'n ontwerpmetodologie vir verspreide databasisse te finaliseer.

1.5.7 FINALISERING VAN 'n METODOLOGIESE RAAMWERK VIR VERSPREIDE DATABASISONTWERP

Ons het in die vorige afdelings verskeie onderwerpe beskou, nl. modelle, fases, benaderings ens. Hierdie onderwerpe moet nou geïntegreer word om 'n geheelbeeld te vorm, wat ons kan gebruik om 'n raamwerk vir 'n ontwerpmetodologie vir verspreide databasisse, te skep. Hiervoor moet ontwerpfasies, die entiteitverwantskap-model, die relasiemodel en verspreide databasis konsepte, in 'n metodologie uitgedruk word wat verspreide databasisontwerp sal kan hanteer. Voordat daarmee voortgegaan word, gaan ons die aspekte betrokke by 'n suksesvolle ontwerpmetodologie beskou.

1.5.7.1 SUKSES-FAKTORE BETROKKE BY 'n ONTWERPSMETODOLOGIE

Alfred F. Case [36] (p3,15) meld in sy artikel "Implimenting CASE", dat die term metodologie wyd-gedefinieerd is, wat kan lei tot probleme in automatisasie van programmatuurontwerp. Ons moet onthou dat hierdie metodologie ook automatiseerbaar behoort te wees. Case verskaf 'n enger definisie oor die term "metodologie", wat soos volg lui:

" 'n Metodologie is 'n stel metodes, prosedures en hulpmiddels wat die proses beskryf waardeur gebruikers se behoeftes gereedlik na werkende rekenaarprogramme, menslike prosedures, operasionele instruksies en onderhouds dokumentasie omgesit kan word sodat die gebruikers se behoeftes bevredig is. "

'n Ontleding van die bg. definisie lei tot die volgende, nl. Metodes is stelle instruksies vir modelleringsprosesse, en data, wat 'n stelsel in terme

van modelle sal voorstel, tesame met model transformasie-algoritmes wat ook behoeftes vir veranderinge sal reflekteer. Case noem entiteitverwantskapmodellering as 'n toepaslike metode. Verder noem hy ook dat 'n metode tipies deur een individu uitgevoer kan word.

Prosedures is instruksies waardeur 'n groep persone wat metodes uitvoer, bestuur kan word in terme van werkseenhede. Prosedures word in terme van diskrete stappe of fases voorgestel.

Hulpmiddels (Tools) stel ontwikkelaars in staat om die metodes en prosedures meer effektief te benut. Tipiese hulpmiddels is redigeerders, vertalers, woordeboeke, rekenaargrafika ens.

Case noem dit dat 'n goed gevormde metodologie wel geïntegreerde metodes, prosedures en hulpmiddels sal bevat, wat beide "Wat-om-te-doen" en "Hoe-om-dit-te-doen", beskryf. Die stappe van die metodologie sal ook aflewerbare produkte kan produseer.

Die metodologiese raamwerk wat ons gaan ontwikkel vir verspreide databasisontwerp, gaan net elemente van metodes bevat, omdat die studiegebied van hierdie verhandeling nie alle besonderhede kan bevat nie (dit is te wyd). 'n Verdere ontwikkeling van die voorgestelde metodologiese raamwerk kan in ander proefskrifte, in meer besonderheid bespreek word.

Ons het reeds die verskillende aspekte betrokke by die ontwerpsfases beskou, nl. die gebruik van datamodellering, ontwerpsfases en benaderings, en verspreide databasisontwerp. Hieruit gaan ons 'n moontlike raamwerk vir 'n ontwerpmetodologie ontwikkel wat die behoeftes van verspreide databasisontwerp, sal kan bevredig.

Die leser moet in gedagte hou dat die voorgestelde raamwerk nie die enigste oplossing vir verspreide databasisontwerp is nie, maar net een van vele moontlikhede is. Die studieveld van verspreide databasisontwerp is nog nie breed genoeg ontwikkel om pertinente oplossings voor te stel nie, maar die hoop word uitgespreek dat hierdie verhandeling 'n rigtinggewende invloed op die studieveld sal kan uitoefen, al is dit uiters gering. Die voorstelle in hierdie verhandeling is dus geheel en al oop vir bespreking.

1.5.7.2 DIE SINTESE VAN 'n METODOLOGIESE RAAMWERK VIR VERSPREIDE DATABASISONTWERP

'n Voorgestelde metodologiese raamwerk, is gesintetiseer uit die gebiede wat in die vorige gedeeltes bespreek is nl. die aspekte betrokke by verspreide databasisontwerp, datamodellering, die ontwerpstrategie en die ontwikkelingsfases. Die basiese raamwerk sal gebou word op die vorige besprekings van ontwerpfases (Gedeeltes 1.5.1 - 'n Ontleding van tradisionele ontwerpfases {p14} en 1.5.3 - Vereenvoudigde ontwerpfases vir verspreide databasisse {p22}).

Die tradisionele ontwerpfases is deur Teorey en Fry [33] beskryf as : (1) Behoeftebepaling en analise, (2) Logiese ontwerp en (3) Fisiese ontwerp en evaluering, maar is aangepas vir verspreide databasisontwerp en die modelleringsbenadering. Ons moet dit ook nooit uit die oog verloor dat ons eintlik met verspreide databasisontwerp besig is nie, d.w.s die verspreidingsveranderlike (geografiese plek) is uiters belangrik in die ontwerpsproses, en moet deurgaans in ag geneem word. Die veranderlike van "PLEK" is die enigste veranderlike wat 'n verspreide databasis uniek identifiseer, in

teenstelling met 'n gesentraliseerde databasis. Die verspreidings-element moet dus pertinent in die metodologie genoem word, en as 'n belangrike analise-onderwerp voorkom.

Die fases vir 'n metodologie vir verspreide databasisontwerp, soos gesintetiseer in afdeling 1.5.3 bestaan uit ontleding, konseptuele aspekte, toepassingsanalise en logies-fisiese aspekte. Dit word dan gekombineer met die unieke eienskap van 'n verspreide databasis (die "PLEK" veranderlike) en datamodellering (entiteitverwantskapmodel en die relasiemodel) sodat die volgende fases voorgestel kan word:

- FASE I Behoeftebepaling en ontleding.
- FASE II Verspreidingsanalise.
- FASE III Konseptuele entiteitverwantskapmodellering.
- FASE IV Logiese toepassing spesifisering.
- FASE V Logies-Fisiese Relasiemodellering.

Bg. voorgestelde fases bevat al die nodige elemente vir die behoeftes van konseptuele ontwerp (Fase III), logiese ontwerp (Fase IV en V), analise van die probleemstelling (Fase I), verspreide databasis (Fase II) en modellering (Fase III en V). Hierdie raamwerk blyk dus om voldoende te wees vir die doeleindes van 'n metodologie vir verspreide databasisontwerp.

Daar is wel metodologië ontwikkel vir gesentraliseerde databasisontwerp wat fisies in gebruik is, en waaruit ons kan leer. Een van die genoemde metodologië is die INCOD stelsel. Atzeni, Batini, Lenzerini en Villanelli [28] (p375) onderskryf die INCOD databasis ontwerpstelsel, en noem dat 'n metodologie moet bestaan uit 'n stel gereedskap, hulpmiddels en tegnieke wat ten doel databasisontwerp, die volgende funksies moet kan verrig:

1. Om gebruiker behoeftes te versamel.
2. Die databasis te kan ontwerp sodat die behoeftes van die gebruiker op 'n effektiewe wyse bevredig kan word deur die inligting-analise en definisie (konseptuele ontwerp) tesame met die implementasie en fisiese ontwerp van die databasis te hanteer.
3. Om beplannings hulpmiddels te genereer.
4. Om dokumentasie te genereer wat nuttig sal wees vir die ontwerp en implementasie sowel as die verdere onderhoudsfases van die databasis.

Hulle noem dit dat die tradisionele twee-vlak benadering (logiese en fisiese ontwerp) te wyd gespesifiseer is. Die oplossing hiervoor is om die metodologie stapsgewys op te breek.

'n Fyner metodologie behoort ook 'n beter oplossing vir verspreide databasisontwerp te wees, omdat daar heelwat ekstra ontwerpsaspekte is, wat in aanmerking geneem moet word. Hierby vind ons bv. databasis heterogeniteit, netwerk ladingsanalise ens. 'n Fyner metodologie sal die ontwerper meerendeels in staat stel om 'n baie komplekse ontwerp te hanteer. Die feit dat die voorgestelde metodologie uit vyf diskrete fases bestaan, kan dus geregverdig word.

Om die voorgestelde metodologiese raamwerk vir verspreide databasisontwerp uit te brei tot 'n volwaardige metodologie, moet hulpmiddels en metodes ingebou word in alle fases, wat die metodologiese behoeftes van Atzeni, Batini, Lenzerini, Villanelli [28] sal bevredig.

Die voorgestelde metodologie word nie volledig bespreek nie, net 'n moontlike raamwerk vir verdere verfyning word verskaf. 'n Verdere uitbreiding van die

fases tot 'n volledige en volwaardige metodologie kan dan deur latere studies gedoen word.

Ons het tot dusvêr 'n raamwerk ontwikkel wat gebruik kan word vir die ontwerp van verspreide databasisse. Al wat oorbly is om die fases van die raamwerk in meer besonderheid te bespreek en aan te toon hoe dit toegepas moet word. Elkeen van die verdere hoofstukke van die verhandeling bespreek die inhoud van een van die fases van die metodologiese raamwerk. Die struktuur van die verhandeling is soos volg:

1.6 DIE STRUKTUUR VAN DIE VERHANDELING

Hierdie verhandeling word verdeel in sewe verskillende hoofstukke. Elke hoofstuk gaan aspekte binne fases van die voorgestelde metodologie vir verspreide databasisontwerp beskryf. Ons het reeds kennis gemaak met 'n paar van die aspekte waarop daar in die volgende hoofstukke uitgebrei sal word, soos die datamodelle wat gebruik kan word. Ons sal ook vind dat 'n gedeelte van elke bespreking in elke hoofstuk ook op sentrale databasisse toegepas kan word, omdat baie van die teorie hoofsaaklik uit sentrale databasisontwerp geneem is. Die sewe hoofstukke en hul onderwerpe is die volgende:

Hoofstuk 1: DIE ONTWIKKELING VAN 'N ONTWERPSMETODIEK DEUR DIE ANALISE VAN BETROKKE ASPEKTE BY DATABASISSE, VERSPREIDING EN DATAMODELLE.

Hierin is die doel van die verhandeling beskryf en die onderwerpe betrokke by die ontwikkeling van 'n ontwerpsmetodiek, bespreek. Dit sluit die ontwerpases, benaderings en datamodellering in. 'n Raamwerk vir 'n ontwerpsmetodiek vir verspreide databasisontwerp word ook voorgestel. Hierdie hoofstuk is dus die

aanknopingspunt vir verdere besprekings. Die volgende hoofstuk hanteer die eerste fase vir die ontwerp van verspreide databasisse.

Hoofstuk 2: FASE I : ORGANISASIE ANALISE - BEHOEFTEBEPALING EN ONTLEDING.

Hoofstuk 2 is 'n bespreking van die eerste fase van die ontwerpmetodiek wat in hoofstuk 1 voorgestel is. In hoofstuk 2 gaan ons probleem analise en behoeftebepaling doen en aantoon hoe die organisasie en die gebruikers se behoeftes op die voorgrond geplaas moet word, sodat ons 'n volledige beeld van die organisasie kan vorm. Ons gebruik die organisasie se organigram as beginpunt vir ontleding, en werk deurgaans die verspreidingsdimensie van die organisasie in. Ons sal ook verdere analises op die struktuur van die organisasie doen, en ook na verbetering van die organisasie kyk. Data-analise word ook op die organisasie toegepas. Die volgende hoofstuk (hoofstuk 3) beskryf die fase vir verdere ontleding van die verspreiding van die organisasie, sodat die verspreidheid van die organisasie vroegtydig behandel word.

Hoofstuk 3: FASE II : VERSPREIDINGSANALISE.

In hierdie hoofstuk beskou ons die tweede fase van die metodologie, nl. 'n analise van die fisiese verspreide struktuur van die organisasie, d.w.s. waar die afdelings in die organisasie fisies funksioneer. Die belangrikste inset vir hierdie fase is die uitvoer van fase I soos bespreek in hoofstuk 2. Ons gaan 'n globale verspreidingsanalise doen om die lokaliteit van data aan te toon. Verder gaan ons 'n affiniteitsanalise op die elemente wat vooraf ontleed is uitvoer, om sodoende die elemente saam te groepeer. Hierdie elemente is die

funksies en entiteite wat vooraf in hoofstuk 2 geïdentifiseer is. Na hierdie fase sal genoeg inligting tot ons beskikking wees om die konseptuele model van die organisasie te vorm. Dit word in hoofstuk 4 gedoen.

Hoofstuk 4: FASE III: KONSEPTUELE - LOGIESE ENTITEITVERWANTSKAPMODELLERING.

Ons gaan 'n uitgebreide entiteitverwantskapmodel gebruik om die data in die verspreide organisasie te modelleer. Ons kan hierdie model ook die konseptuele model noem. Die resultaat van hierdie fase is 'n EV-model wat 'n konseptuele en logiese beeld van die organisasie sal toon. Ons kyk ook vlugtig na die moontlikheid van transformasie vanaf die EV-model na die relasiemodel d.w.s. vanaf 'n konseptuele model na 'n fisies implementeerbare model. Daar moet onthou word dat die EV-model gebruiker geïntereerd is, en hoofsaaklik as 'n kommunikasie- en ontwerpmodel gebruik word, om spesifisering van die datamodel te verbeter. In die volgende fase (hoofstuk 5 - fase IV) kyk ons na die funksies van die databasis oftewel die toepassings.

Hoofstuk 5: FASE IV : LOGIESE TOEPASSING SPESIFISERING.

In hierdie hoofstuk gaan ons spesifiek kyk na die funksies wat vooraf geïdentifiseer was en dit in deel in prosesse wat getransformeer kan word na rekenaarprogramme of handstelsels. Ons gebruik hoofsaaklik die bo-na-onder benadering vir die identifisering van die prosesse. Die vooraf geïdentifiseerde funksies (uit fase I en II) sal gebruik word as die ontledingsbron. Die prosesse kan bv. in terme van EV-model gebaseerde datamanipuleerstellings gedefinieer word. Hierdie fase word ook gebruik as inset vir die fragmentasie en plasing van relasies wat in die volgende fase, (fase V - logiese-fisiese

relasiemodellering) gedoen word. Die volgende fase hanteer die fisiese implementeerbare databasismodel d.w.s. die relasiemodel.

Hoofstuk 6: FASE V : LOGIES-FISIESE RELASIEMODELLERING.

In hierdie fase word die transformasie vanaf die EV-model na 'n bruikbare fisiese implementeerbare model nl. die relasiemodel behandel. Ons skep dus 'n logiese model wat rekenaar georiënteerd is. Daar moet ook in ag geneem word dat verspreidingsanalise (fase II) en logiese toepassing spesifisering (fase IV) gebruik gaan word om plasing en fragmentasie van die databasis te definieer. Die plasing- en fragmentasie-aspekte word beskryf as die fisiese gedeelte van die relasiemodellering. Die relasiemodel word gegenereer vanaf die EV-model deur 'n transformasie algoritme te gebruik. Daar moet tog 'n mate van handgedrewe inset gelewer word om te verseker dat die model optimaal opgestel word. Hierdie fase gaan ook die implementering van die databasis vooraf en is die laaste stap van die ontwerpmetodiek.

Hoofstuk 7: AFSLUITING EN GEVOLGTREKKING.

In hierdie hoofstuk gaan ons die totaliteit van die teorie wat bespreek is beskou, en 'n gevolgtrekking maak wat 'n veralgemening sal insluit oor die bevindinge in die vorige hoofstukke.

1.7 OPSOMMING

In hoofstuk 1 het ons gekyk na die rol van verspreide inligting in die organisasie, en gevind dat inligting 'n baie belangrike rol speel as hulpbron in enige organisasie. Verder

het ons gekyk na die voordele wat verspreide databasisse inhou, in terme van kostebesparings, die struktuur van die organisasie, en aanpasbaarheid tot verandering. Daar is gevind dat verspreide databasisse baie voordele vir 'n organisasie inhou as die struktuur en behoeftes van die organisasie dit regverdig.

Daarna het ons voortgegaan om 'n metodologie te ontwikkel waarmee ons verspreide databasisse kan ontwerp, deur te kyk na ontwerpsfases, strategië, datamodelle en metodologiese aspekte. Daar was gevind dat 'n metodologie vir verspreide databasisontwerp redelik fyn ontwikkel moet wees en uit fases vir behoeftebepaling, verspreidingsanalise, konseptuele-logiese modellering, logiese toepassing spesifisering en logiese-fisiese modellering moet bestaan. Die volgende hoofstuk behandel die eerste ontwerpfase nl. Fase I : Organisasie analise - behoeftebepaling en ontleding.

HOOFSTUK 2

FASE I : ORGANISASIE ANALISE - BEHOEFTEBEPALING EN ONTLEDING.

Opsomming

Hierdie hoofstuk is 'n oorsig oor die onderwerpe wat betrokke is by behoeftebepaling en organisatoriese ontleding betrokke by verspreide databasisse. Ons kyk onder andere na die inligtingvloei in die organisasie, bronne van data, identifisering van funksies en die ontleding van die organigram. Die doel van hierdie fase is om seker te maak dat ons die verspreide databasis op 'n geoptimaliseerde organisasie sal toepas, en dat ons, ons self sal vergewis van presies hoe die organisasie funksioneer. Die basiese uitset van hierdie fase is 'n matriks van die organisasie se funksies en data-objekte, tesame met 'n organigram van voorgestelde verbeterings aan die organisasiestruktuur.

2.1 INLEIDING

Een van die belangrikste redes vir die belangstelling in verspreide databasisse, ontstaan uit die behoefte van verspreide organisasies om verwerkingskrag te versprei (Oxborrow [38] [p89]). In fase I word die klem geplaas op die verspreide organisasie en die gebruiker se behoefte vir 'n verspreide databasis. Ons ontledings onderwerp is dus die verspreide organisasie en die werking daarvan.

Ook kyk ons na die huidige struktuur van die organisasie met die oog op verbeteringe sodat ons die databasis op 'n optimale organisasiestruktuur kan toepas. Dit sal nie help om die beste ontwerpte verspreide databasis toe te pas op 'n oneffektiewe en onderbenutte organisasiestruktuur nie. Tschritzis en Lochovsky [9] noem dat 'n hoëvlak ontleding van die werking van die organisasie gedoen moet word. Ons kyk ook na 'n formele wyse van ontleding, (data-analise) wat op die

modellerings-benadering toegepas kan word, d.w.s. die resultaat van hierdie fase moet maklik omskep kan word in modellerings terme. Dit moet nogtans ook die gebruiker se behoefte vir 'n verspreide stelsel reflekteer. Die uiteindelige doel is om seker te maak dat ons 'n geïntegreerde inligtingstelsel vir 'n effektiewe en optimale organisasie, sal kan implementeer.

Hierdie fase se uitset is 'n geoptimaliseerde organisatoriese model, wat kan dien as inset vir verdere ontwerpsprosesse. Hierdie model is uiters kritiek omdat dit die implementasie sukses van die verspreide databasis mag beïnvloed. Ons moet dus eerstens uitvind hoe die organisasie funksioneer, en waar daar moontlike probleme kan wees, voordat ons 'n verspreide databasis vir daardie organisasie kan ontwerp. Daarna moet ons probeer om enige bottelnekke en oneffektiewe eenhede te diagnoseer, sodat ons die organisatoriese model kan verbeter. Die resultaat van hierdie fase beskryf die organisasie in terme van data-objekte, funksionele aktiwiteite en die organigram, en nie in terme van enige formele datamodel nie. Datamodelle wat die datastruktuur van die organisasie beskryf, sal in die latere fases (hoofstukke 4 en 6) ontwikkel word.

In hierdie hoofstuk vergewis ons, ons self van die werking van die totale organisasie en probeer ons dit verbeter sodat die databasis op 'n optimale organisasie toegepas kan word. As dit nie gedoen word nie, moet ons die oneffektiewe werking van sekere gedeeltes van die organisasie aanvaar, en die databasis binne daardie oneffektiewe raamwerk implementeer.

Ons gaan in die volgende gedeelte die ontledingsproses beskou, d.w.s. die organisasie as 'n geheel ontleed en betrokke faktore uitlig.

2.2 ONTLEDINGFASES

Die ontledingsproses begin met 'n ondersoek van die doelstellings van die organisasie, en die voordele wat die implementering van 'n verspreide databasis sal meebring. Die nuttige rol wat inligting in die organisasie speel, moet heelyd in gedagte gehou word. Verwys gerus terug na paragraaf 1.2 (hoofstuk 1 - p4). Om die ontledingsproses aan die gang te kry deur die doelstellings van die organisasie te ondersoek, kan ons eksterne konsultante inroep om besigheidsontledings van die struktuur van die organisasie te doen. Dit kan in samewerking met die organisasie se interne stelselontleders gedoen word. Martin [2] (p286) noem dat die gebruik van eksterne konsultante die moontlikheid van subjektiewe ontledings sal uitskakel. Ons gebruik 'n twee-fase ontledingsproses wat eerstens die huidige stand van sake analiseer, en daarna moontlike verbeterings in die bestaande organisasie aanbring.

Die eerste gedeelte van die ontledingsproses se resultaat, is 'n dokument wat die doelstellings van die organisasie beskryf, en die impak van 'n verspreide databasis daarop sal uitwys. Ons maak dus eerstens seker dat die implementering van 'n verspreide databasis die beste keuse is, in teenstelling met die implementering van 'n gesentraliseerde databasis. Nadat hierdie rigtinggewende dokument ontwikkel is, moet 'n data-analise van die organisasie gedoen word. Dit is nog deel van die eerste fase in die ontleding van die organisasie. Daarna sal daar oorgegaan word om datamodelle te gebruik om sodoende formalisasie van ontwerp in die ontledingsproses aan te vul.

Dit is belangrik dat die modelle wat verder in die ontledingsproses gekies word, ook deur die persone binne die organisasie verstaan word (Tsichritzis en Lochovsky [9] (p 252)). In hierdie hoofstuk kyk ons nie na 'n spesifieke organisatoriese voorstellingsmodel nie, maar beskryf ons net die basiese elemente waaruit so 'n model bestaan, nl. entiteite en funksies. Die keuse en voorstelling van 'n model

in hierdie fase, word aan die gebruiker van die metodiek oorgelaat. Ons gebruik 'n verwysing na die organisasie se organigram, wat 'n model van die struktuur van die organisasie is.

Die ontledingsproses vir fase I bestaan uit die volgende:

1. Doelstelling en behoefte analise.
2. Organisasoriese analise.
3. Data-analise.
4. Verbeterde organisasoriese model.

Ons uitset vanaf hierdie ontledingsproses is 'n regverdiging vir die implementering van 'n verspreide databasis, funksionele en entiteit identifikasies in terme van 'n matriks, en 'n voorgestelde verbeterde organigram. Daar moet in gedagte gehou word dat 'n implementeringsplan vir die verbeterde organisasie tesame met die ontwerp van die verspreide databasis gedoen moet word, sodat die databasis en verbeterings tegelyk geïmplementeer kan word. Ons gaan nou kyk na inligtingvloei in die organisasie sodat ons kan uitvind waar die grootste impak van 'n databasis sal wees. Ons kyk dus na hoe 'n databasis die horisontale en vertikale vloei van inligting sal beïnvloed. Daarna kyk ons na die bronne van inligting.

2.3 DIE VERSPREIDE DATABASIS SE INPAK OP INLIGTINGVLOEI

'n Doenlikheidstudie wat die impak van die stelsel op die organisasie sal toon, moet eerstens aangespreek word. Dit pas dus in met wat ons voorheen genoem het, nl. " is die implementering van 'n verspreide databasis geregverdig?" Hou in gedagte dat ons in terme van hierdie verhandeling net na verspreide databasisse kyk. Hierdie ontleding word in terme van die organisasie se vertikale-en-horisontale inligtingvloei uitgedruk.

Die vertikale vloei bestaan uit direktiewes (bo-na-onder) en inligting (onder-na-bo). Die direktiewes se detail vermeerder van bo-na-onder terwyl inligting van onder-na-bo vermeerder d.w.s. data word in die proses omgeskakel in inligting. Ons kan dit ook as sterk hiërargies beskou.

Horisontale inligtingvloei is bloot 'n uitruiling van data tussen interafhanklike funksionele eenhede binne die organisasie. Hierdie horisontale vloei impliseer 'n groot mate van data-deling tussen die outonome eenhede in die organisasie. Ons moet dus eerstens verseker dat die organisasie se doelwitte bevredig kan word deur om lang- en medium-termyn projeksies met die dag na dag onder-na-bo datavloei te kombineer. Ons doel is om 'n geïntegreerde inligtingstelsel te implementeer en om die inligtingvloei in die organisasie te verbeter. Die inligtingvloei kan geïllustreer word deur die organigram te gebruik om die vloei moontlikhede aan te toon.

Nadat die doenlikheidsstudie vir die huidige organisasie voltooi is, en daar bv. gevind is dat 'n verspreide databasis nie die antwoord is nie, moet ons nogtans voortgaan met die analise totdat ons uiteindelik 'n verbeterde organisasie-model tot gevolg het. Daarna moet ons weer dieselde studie doen en mag ons moontlik vind dat 'n verspreide databasis nou wel definitiewe voordele inhou. Die omgekeerde kan ook waar wees. Ons probeer dus in hierdie fase uitvind of 'n verspreide databasis wel die antwoord op die organisasie se probleme en behoeftes is, deur inligtingvloei te analiseer en 'n doenlikheidsstudie van die impak van die databasis op die organisasie te doen.

Ons behoort ook uit te vind waar inligting in terme van die organisasie bekom kan word. Die volgende gedeelte bespreek die bronne van inligting binne die organisasie. Die stelselontwerpers kan dus hierdie bronne gebruik om inligting i.v.m. die organisasie te bekom.

2.4 INLIGTINGSBRONNE IN DIE VERSPREIDE ORGANISASIE

Ons sien uit die voorafgaande gedeelte dat ons die horisontale en vertikale inligtingvloei in die organisasie moet benut om ons analise en behoeftebepaling te doen, en om die impak van die verspreide databasis te ondersoek. Verder moet ons inligting inwin i.v.m. die organisasie. Die vernaamste bronne van inligting is mense en bestaande dokumentasie (Tsichritzis en Lochovsky [9] (p251)). Martin [2] ondersteun ook hierdie stelling. Hulle is verder van mening dat die inligtingsbehoefes van die organisasie by gebruikers op alle vlakke binne die organisasie ingewin moet word (vanaf topbestuur tot by die produksievloer).

Vir doelgerigte analise van inligting moet iteratiewe onderhoude en dokumentontleding gevolg word. Iterasie is belangrik om konflikte op verskillende vlakke en tussen verskillende gebruikers op te los sodat meer insig a.g.v. die onderhoude en opleeswerk in die organisasie verkry kan word.

Die vernaamste menslike inligtingsbronne in 'n organisasie kan in 3 vlakke verdeel word nl. topbestuur, middelbestuur en operasionele-bestuur. Hulle is die vernaamste bron van metadata (data wat data beskryf), omdat hulle alle horisontale en vertikale inligtingvloei dek.

Topbestuur word gebruik om die doelstellings en strategië van die organisasie te verkry. Verder kan hulle ook die bestuurs-aspekte vir die bereiking van die doelstellings identifiseer. Ons moet hulle ook gebruik om 'n toekomsgerigte beeld van die organisasie te skep, (lang en medium termyn) en om die middelbestuur te identifiseer. Martin [2] noem dat rekenaarpersoneel grootliks probleme ondervind rakende kommunikasie met topbestuur, omdat hulle uitgangspunt te tegnies van aard is. Martin meen dat topbestuur benader moet word bloot deur te sinspeel op verbeteringe in die organisasie, en nie op die tegniese aspekte van die verspreide databasis nie. Die doel van enige rekenariseringsprojek is

juis om die organisasie te verbeter. Topbestuur moet ook vooraf daarvan bewus gemaak word dat die huidige verspreide organisasie-model moontlik sal moet verander, om 'n meer effektiewe organisasie-struktuur op die verspreide databasis toe te pas. Hierdie verandering moet met die heelhartige samewerking en ondersteuning van die topbestuur ontwikkel word.

Middelbestuur verskaf metadata in meer besonderheid, wat die omvang van die analise en ontwerp sal beskryf. Hulle verskaf ook praktiese persepsies i.v.m. die middel- en kort-termyn bedryf van die organisasie. Hulle identifiseer ook fisiese aspekte soos sekerheid, responstye, beheer, bestuurstake en besigheidsfunksies, asook om operasionele bestuurspersoneel aan te dui. Ons vind ook dat hulle in 'n groter mate van tegniese betrokkenheid by die stelsel belangstel.

Operasionele bestuur verskaf inligting i.v.m. spesifieke data-elemente en laevlak fisiese aangeleenthede soos die gebruik van data, volumes, gebruiksfrekwensies, transaksies en prioriteite, ordening en prestasie kriteria. Hulle is dus ons basisvlak-bron van struktureeringsdata vir die databasis implementering.

Ons moet ook onthou dat ons in die volgende fases deurgaans sal terugverwys na die menslike inligtingsbronne om ons ontwerp te verfyn. Die iterasie beginsel kom dus deurentyd voor.

Die vernaamste dokument wat ons kan gebruik vir die identifisering van die organisasie-struktuur en vlakke, is die organisiekaart of organigram. Dit is 'n diagram wat die hiërargiese vlakke in die organisasie in terme van poste, afdelings en mense weergee. Ons kan hierdie dokument gebruik as aanknopingspunt vir behoeftebepaling en ontleding omdat dit 'n reflektoring van die werklike struktuur van die organisasie is. Martin [2] noem dat dit belangrik is om die organigram te rekenariseer omdat dit dan meer manipuleerbaar

sal wees. Alle ander dokumente, soos verskaf deur die persone in die organisasie, moet ook ontleed, gebruik en geliaseer word sodat ons daarna kan terug verwys as dit later nodig word.

Ons het tot dusver na die ontledingsproses gekyk, en die kommunikasievlakke en bronne van inligting ontleed, d.w.s waar die inligting te vinde is. Ons beweeg nou in die rigting van 'n formele term vir die ontleding van die organisastoriese inligting tot ons beskikking, d.w.s hoe die inligting ontleed word. Dit word "data-analise" genoem en sal volgende behandel word.

2.5 DATA-ANALISE

Davenport [35] meld dat ontleding van verspreide organisasies se behoeftes, onder die vaandel van data-analise geplaas moet word. Hy definieer dit in 'n bespreking gedokumenteer in die Infotech State of the Art Report, DISTRIBUTED DATABASES - Volume 1: Analysis & Bibliography [34] soos volg:

"The objective of Data Analysis is to determine the fundamental nature of an organisation's data resources and to organise and document all relevant facts concerning this data and thus provide a sound basis for database design."

Hy meen dat data-analise uit twee stappe bestaan nl. a) entiteit-analise en b) funksionele-analise. Entiteit-analise verskaf 'n metode wat gebruik kan word om 'n komplekse omgewing te verstaan en te dokumenteer, in terme van die entiteite wat daarin bestaan. Funksionele-analise is die dokumentasie en analise van die basiese besigheidsaktiwiteite in die organisasie.

Martin [2] gebruik ook funksionele-analise en entiteit-analise

om 'n oorsigtelike model van die organisasie te ontwikkel. Ons sien dus dat data-analise (op 'n hoë vlak) van toepassing is op hierdie fase van die ontwerpmetodiek omdat ons dit kan gebruik om die struktuur van die organisasie te ontleed. Ons gaan verder in meer besonderheid kyk na die elemente van data-analise nl. funksionele-analise en entiteit-analise.

2.5.1 FUNKSIONELE-ANALISE

Die operasionele werking van die organisasie kan in terme van funksies voorgestel word. Tsichritzis en Lochovsky [9] (p 252) definieer 'n funksie as 'n essensiële aktiwiteit of besluit, wat benodig is om die hulpbronne en die bedryf van hulpbronne binne die organisasie te beheer (proses of prosedure). Funksies word geïdentifiseer deur die volgende te ondersoek: beskrywings van die doel van take, werkprogramme binne organisasie areas en deur produkte en dienste binne die organisasie te ontleed in terme van die funksies wat gebruik moet word om dit te lewer.

Die funksies van 'n organisasie definieer die aktiwiteite wat die organisasie moet verrig om die doelstellinge van die organisasie te vervul. Dit is belangrik om werknemers se funksies op verskillende vlakke van die organigram te analiseer en te ontleed, omdat dit 'n volledige beeld van die bedryf van die organisasie sal skep. Voorbeelde van organisatoriese funksies is bemaking, produksie en berekening. Funksies vir personeelgroepe en individuele personeel moet ook gevind word, bv. funksies van die onderneming, funksies van topbestuur, funksies van middelbestuur, funksies van die personeelbeampte ens.

Martin [2] (p325) definieer 'n besigheidsfunksie as: 'n groep aktiwiteite wat die doelstellings van die onderneming bevorder. Hierdie besigheidsfunksies kan weer

in funksionele areas ingedeel word. Die beginpunt van funksionele-analise is die identifisering van die funksionele areas, d.w.s die identifiseerbare werk wat die organisasie verrig, word gegroepeer om hulle missie te bereik. Daarna kan elke funksionele area in die besigheidsfunksies opgedeel word. Hy tref onderskeid tussen 'n besigheidsfunksie en 'n proses. 'n Besigheidsfunksie kan verder in prosesse verdeel word.

'n Proses word gedefinieer as: 'n aksie met 'n definitiewe begin en einde (p328). Proses identifikasie is eers in latere fases (fase IV) van die ontwerpmetodiek belangrik, omdat ons in fase I die organisasie net op 'n hoëvlak beskou. Die funksies sal dan later ontaard in die toepassings wat op die databasis uitgevoer gaan word. Die funksies is van toepassing op die entiteite in die organisasie. Die volgende gedeelte behandel entiteit-analise.

2.5.2 ENTITEIT-ANALISE

Davenport [35] (p95) verskaf sewe stappe wat gevolg kan word, waarmee entiteit-analise gedoen kan word, alhoewel hierdie stappe op 'n laer vlak gedefinieer word, as wat in hierdie fase nodig is. Martin [2] gebruik 'n hoëvlak entiteit diagram as inset na die organisasie model. Hy identifiseer eerstens "data-onderwerpe" wat later vereenvoudig word in terme van entiteite. Sulke data-onderwerpe is bv. "VERKOPE AREAS", "KLIËNTE", "BESTELLINGS", "PRODUKTE", "ONDERDELE", "MATERIAAL", "VERKOOPS EINDPUNTE", "WERKNEMERS", "TOBRUSTING" ens., dus oorhoofse groeperings van die organisatoriese onderwerpe waarvan data gehou kan word, en wat uit entiteite bestaan. Vir elke data-onderwerp word dan ook entiteit-tipes in besonderheid geïdentifiseer. Martin identifiseer 'n entiteit as : enige persoon of ding,

werklik of abstrak waarvan inligting gehou kan word bv. vir data-onderwerp "BESTELLING", kan die entiteit-tipes "KLIËNT", "ITEM", "BETALING" ens. geïdentifiseer word.

Tsichritzis en Lochovsky [9] (p 252-253) verskaf 'n variasie op die "data-onderwerp" beginsel, soos deur Martin [2] genoem, naamlik "dataklasse". 'n Dataklas is afhanklik van die bestaan van 'n funksie. Dataklasse kan dus eers geïdentifiseer word nadat funksionele analise gedoen is. 'n Dataklas word gedefinieer as : 'n versameling van data attributes wat benodig word deur 'n funksie. Dataklasse en data-onderwerpe is net 'n variasie op die tema van data-analise en kan beide met ewe veel vrag gebruik word.

Die keuse van die implementering van entiteit-analise is afhanklik van die gebruiker van die metodologie. Die uiteindelige doel is om entiteite te identifiseer waaruit die organisasie opgebou is, en daarna 'n model van die verspreide organisasie op te stel.

Die entiteite en funksies kan in terme van 'n matriks voorgestel word, sodat 'n beter beeld van die voorstelling verkry kan word. Dit word in die volgende gedeelte bespreek.

2.5.3 MATRIKS-VOORSTELLING

Martin [2] (p340) en Tsichritzis en Lochovsky [9] (p253) gebruik albei tweedimensionele matrikse om funksies teenoor entiteite voor te stel. Dit is 'n uitstekende metode om die verwantskappe tussen die funksies en die data-entiteite in 'n verspreide organisasie aan te toon, sodat ons 'n goeie beeld van die werking van die organisasie, en wat of wie daarby betrokke is, kan vind. Die klasse, wat gebruik kan word

om die voorstellings te doen, is bv. poste of afdelings in die organisasie. 'n Tipiese uitset van fase I sal dan matriksvoorstellings van al die organisatoriese elemente wat ondersoek was, insluit. 'n Voorbeeld hiervan sal later getoon word. Ons gaan nou kyk na die faktore betrokke by die implementering van verbeterings op die verspreide organisasie, deur op die huidige struktuur voort te bou.

2.6 ONTLEDING VAN DIE HUIDIGE VERSPREIDE ORGANISASIE

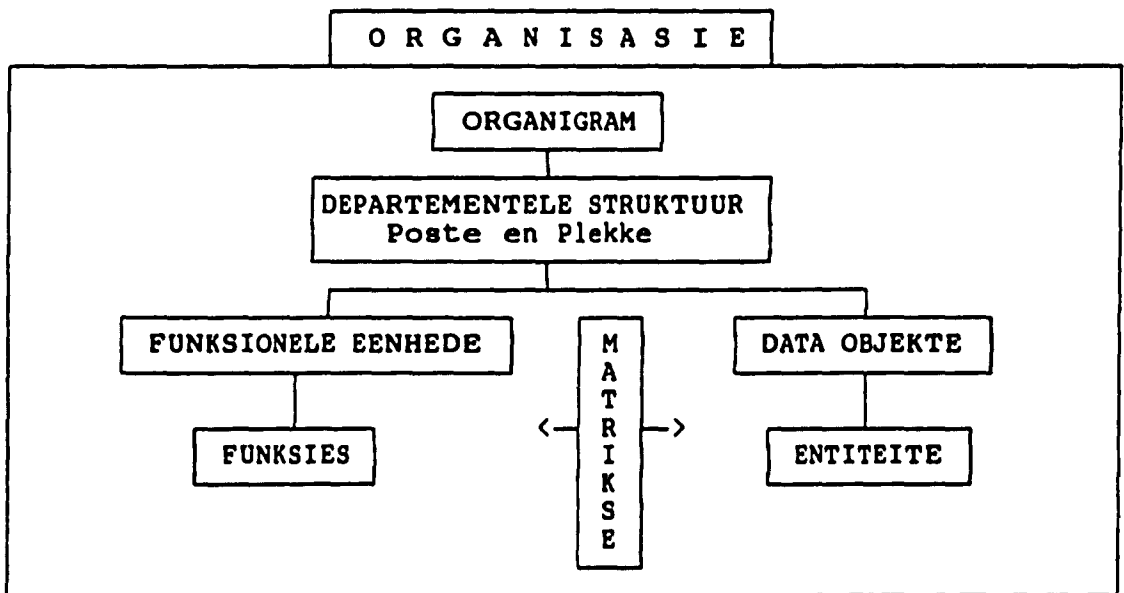
Op die hoogste vlak van analise, gebruik ons die organigram van die organisasie om die departementele struktuur te vind (Martin [2] [p321]). Ons kan bv. 'n verkoopsdepartement, produkontwikkeling, finansiële departement, die verskillende poste in elk, en die verskillende plekke waar dit voorkom, identifiseer. Let op dat plekke pertinent na verspreiding verwys. Daarna doen ons doenlikheidstudies en data-analise op die inligting. Die huidige struktuur van 'n verspreide organisasie kan tipies soos volg ontleed word:

Ons kan funksionele areas direk uit die departementele struktuur van die organisasie identifiseer. 'n Funksionele area word eng aan die dag na dag bestaan van die departement gekoppel. 'n Funksionele area beskryf die werk wat elke departement verrig. Deur stapsgewyse verfyning te gebruik, kan ons elke funksionele area opdeel in hulle onderskeie funksies, d.w.s watter funksies deur elke funksionele area verrig word. Daar moet in gedagte gehou word, dat 'n funksie net beskryf wat gedoen word, en nie hoe dit gedoen word nie.

Later kan ons die funksie in prosesse d.w.s hoe die funksie gedoen word, opbreek. Die proses-vlak is die laagste vlak van analise wat nodig is sodat dit in rekenaarprogramme omgesit kan word. Met elke aspek van die ontleding moet die plek van voorkoms en kommunikasieverbindings met ander plekke ook gespesifiseer word.

Net soos die funksionele areas vanuit die departementele struktuur verkry was, kan ons "data-onderwerpe" identifiseer wat later in entiteite omgesit kan word. 'n Data-onderwerp is 'n oorhoofse groepering van data wat in 'n departement hanteer word om die departement se werk te verrig. Hierdie data-onderwerpe kan nou deur stapsgewyse verfyning onderverdeel word in entiteite of data-items. 'n Entiteit of data-item beskryf enige persoon of ding waarvoor data gestoor kan word. Die plekke waar die entiteite voorkom, moet ook gespesifiseer word.

Ons beskik nou oor 'n lys van departemente, poste en plekke wat in terme van funksionele eenhede en data objekte uitgedruk kan word. Ons kan nou 'n beeld vorm van die funksies en entiteite wat betrokke is by die werking van die organisasie. Skematies kan ons die huidige struktuur van die verspreide organisasie soos volg voorstel:



Ons moet in gedagte hou dat bg. analise net van toepassing is op die huidige organisasie. Ons het dus 'n groot hoeveelheid werk gedoen, om uit te vind hoe die organisasie huidiglik lyk en funksioneer. Die implementering van 'n verspreide databasis:

sal ook moontlike veranderings meebring aan die huidige struktuur van die organisasie. Ons sal nie presies weet wat hierdie veranderings is nie, maar ons kan wel aannames in verband daarmee maak, bv. verandering van kommunikasie-strukture. Hierdie veranderings behoort in 'n verbeterde model van die organisasie gereflekteer te word.

Ons moet ook deurgaans in gedagte hou dat enige organisasie nooit staties van aard is nie, en heeltyd verander. 'n Huidige analise is dus nie van veel nut as ons die volgende paar jaar ook in ag moet neem nie. Daarom moet ons eerstens besluit of die huidige struktuur verbeter kan word, deur na al die logistiese aspekte te kyk en 'n verbeterde struktuur op te stel deur voort te bou op die analise wat reeds gedoen is. Die resultaat is dus 'n verbeterde struktuur van die organisasie. Hierdie aspek word in die volgende gedeelte verder bespreek.

2.7 VERBETERING VAN DIE VERSPREIDE ORGANISASIE EN TOEKOMSBLIK

As gevolg van die koste verbonde aan die implementering van 'n verspreide databasis, moet ons verseker dat ons alle veranderlikes wat kan lei tot die mislukking van die databasis, kan uitskakel. 'n Voorbeeld van so 'n veranderlike is die moontlike effektiwiteit van 'n verspreide organisasie. 'n Verspreide databasis sal gewoonlik net op 'n groot en verspreide organisasie toegepas word, daarom sal dit gewoonlik 'n uiters duur en kritieke implementasie wees.

Dit moet verseker word dat die databasis wel investeringsvoordele sal inhou vir die verspreide organisasie. Ons kan dus nie 'n oneffektiewe en lomp organisasie gebruik as om 'n verspreide databasis op te bou nie, omdat dit die ontwerp en effektiwiteit van die implementering sal benadeel. Die gevolg hiervan kan 'n lomper en nog minder effektiwe organisasie wees. Ons moet verseker dat die organisasie se struktuur en werking, huidiglik, en vir die toekoms, geoptimaliseer sal

wees, om te verseker dat die nut van 'n verspreide databasis tevoorskyn tree in die implementering nl. makliker aanpassing tot verandering.

Die bestuur van die onderneming moet besin oor die effektiwiteit van die huidige organisasie nadat die volgende gedoen is :

- o Behoeftes van die organisasie bepaal is
- o Huidige organisasiestruktuur en werking ontleed is
- o 'n Voorstellingsmodel (organigram) en voorstellingsmatrikse (funksies teenoor entiteite) geskep is.

Die struktuurveranderinge en voordele wat uit die implementasie van die verspreide databasis sal voortspruit, moet in 'n verbeterde organisatoriese model gereflekteer word. Dit is kritiek om die samewerking van die topbestuur hierby te verkry. Die verbeterings kan gedoen word in terme van 'n statiese organisasie (huidige) en 'n dinamiese organisasie (tydsgebonde). Die statiese organisasie neem die huidige model en verbeter waar moontlik, bv. verminder kommunikasie deur oorbodigheid uit te skakel, diversifiseer funksies of benut inligting beter. Die struktuurveranderinge a.g.v. die implementasie van 'n verspreide databasis moet ook spesifiek aangetoon word.

Daar moet na die organisasie gekyk word oor 'n tydperk van 2 jaar, 5 jaar of 10 jaar vanaf die huidige model, en modelle moet vir daardie tydperke opgestel word. Die bron hiervoor moet die verbeterde statiese organisasie-model wees.

Dit is belangrik om op te let dat die modelle, diagramme en kosepte wat gebruik word vir die ontledings, verstaanbaar is vir beide die ontwerpers en gebruikers (bestuur en werknemers van die onderneming). Die verbeterings van die huidige model moet in samewerking met die betrokke bestuur gedoen word. Die bestuur van die organisasie moet dus vasbeslote wees om die organisasiestruktuur te verbeter.

Ons kan verbeterings inisier deur moontlike toekomstige probleme uit te wys. 'n Strategie vir die voortbestaan en groei van die organisasie, wat in terme van 'n implimentasieplan uitgedruk moet word, moet opgestel word.

Dit moet in gedagte gehou word dat die huidige model, verbeterde model, 2 jaar model en 5 jaar model in besigheidsterme uitgedruk moet word waar nodig. Dit is wel moontlik om Chen se EV-model hiervoor te gebruik omdat dit maklik verstaanbaar is, en die komplekse struktuur van 'n organisasie kan reflekteer.

2.8 OPSOMMING

Hierdie hoofstuk handel oor die eerste fase van die ontwerpmetodologie waarin behoeftebepaling en ontleding van die organisasie gedoen word, en waarin gekyk word of 'n verspreide databasis wel die oplossing vir die verspreide organisasie se inligtingsbehoefte is.

Ons neem kennis van die organisasie se funksionering, deur die verskillende bronne van inligting in die organisasie te gebruik, en die impak van die databasis op die verspreide organisasie te ontleed. Ons ontleed ook die impak van die verspreide databasis op die vertikale en die horisontale inligtingvloei in die organisasie.

Ons gebruik hoofsaaklik die organigram en data-analise, (funksie-analise, entiteit-analise en voorstellingsmatrikse) om 'n beeld van die organisasie te vorm. Ons identifiseer verbeterings wat aan die organisasie se struktuur gedoen moet word, en neem die invloed van die verspreide databasis in ag. Verder ontleed ons die organisasie in terme van dinamiese verandering, en probeer ons om doelgerigte tydsgebonde modelle, (2 jaar, 5 jaar en 10 jaar) en implementasieplanne te skep.

'n Finaal ontwikkelde metodologie vir fase I behoort die volgende stappe in te sluit:

- Stap 1: Analiseer huidige organigram en struktuur.
- Stap 2: Ontleed kommunikasievlakke.
- Stap 3: Doen hoë vlak data-analise.
- Stap 4: Stel huidige model op.
- Stap 5: Ontleed inpak van verspreide databasis.
- Stap 6: Stel verbeterde huidige model op.
- Stap 7: Ontleed 2, 5, 10 jaar dinamiese vooruitskouing.
- Stap 8: Stel dinamiese vooruitskouingsmodelle en implementeringsplanne op.

As die voorgestelde stappe gevolg word in hierdie fase, word die volgende uitsette gelewer:

Uitset 1: Organigram van huidige statiese organisasie, verbeterde huidige statiese organisasie (reflekteer die implementering van die verspreide databasis) en organigramme van die dinamiese verbeterde 2, 5 en 10 jaar vooruitskouings van die organisasie. Plekke en kommunikasiekanale vir elke tydperk moet duidelik getoon word.

Uitset 2: Data-analise matrikse vir bg. tydsgebonde organisasies waarop die plekke van voorkoms ook getoon word.

Uitset 3: Implementeringsplanne vir die dinamiese verbeterde organisasie van die huidige model, 2- en 5 jaar modelle. Dit moet in besigheidsterme gespesifiseer word.

N.B. Net die uitsette wat die huidige verbeterde model moet as verdere insette tot die volgende fases van verspreide databasisontwerp gebruik word. Die 2- en 5 jaar uitsette sal later gebruik word om veranderings met betrekking tot die databasis te implementeer.

HOOFSTUK 3

FASE II : VERSPREIDINGSANALISE

Opsomming

Hierdie hoofstuk is 'n oorsig van die onderwerpe wat betrokke is by die ontleding van die fisiese verspreiding van die organisasie, d.w.s die geografiese verspreiding. Die belangrikste inset vir hierdie fase is die uitsette van fase I, nl. die verbeterde organigram en data-analise matrikse. 'n Globale verspreidingsanalise word uitvoer om die plekke waar die entiteite en funksies (uit die data-analise) voorkom, aan te toon. Die uitset van hierdie fase is 'n analitiese tabellering van die fisiese struktuur van die organisasie in terme van verspreiding. Verspreiding is een van die belangrikste en uniekste veranderlikes van verspreide databasisontwerp. Ons ontleed die verspreiding van die organisasie as deel van die ontwerp van die verspreide databasis. Die uitset van hierdie fase is 'n verspreidingsmatriks en affiniteitsanalise en groepering van die elemente binne die matriks. Hierdie uitsette word later in ander fases van die metodologie gebruik.

3.1 INLEIDING

In hierdie fase word die klem geplaas op verspreidingsanalise, d.w.s 'n analise van die fisiese struktuur van die organisasie. Ons gaan die data-analise, (in fase I gedoen) ontleed in terme van die elemente van die organisasie se fisiese geografiese verspreiding. Dit sal gevolg word met affiniteitsanalise wat gebruik kan word vir die fisiese fragmentering en replisering van die databasis. Affiniteitsanalise verskaf ook 'n numeriese en minder subjektiewe metode om verspreiding te ontleed. Analises binne

hierdie fase word aan landkaarte of fisiese plekke gekoppel. James Martin [2] noem 'n belangrike benadering : die skikking en groepering van entiteite deur affiniteitsanalise op die entiteite en funksies toe te pas. Ons gaan hierdie metode gebruik om die inligting wat in fase I versamel is, te groepeer. Hierdie inligting bevat die funksies en entiteite in die organisasie, wat spruit uit die organigram en departementele analise van die organisasie.

Die klem val dus hier op die verspreiding van die organisasie. Omdat ons ten doel het om 'n verspreide databasis te ontwerp, moet ons so vroegtydig moontlik die verspreidings-veranderlike identifiseer sodat ons dit deurgaans in gedagte kan hou in die opeenvolgende ontwerpfasies. Die enigste element wat 'n verspreide databasis anders maak as 'n gesentraliseerde databasis, is dat die afdelings van die organisasie en die meegaande data, by verskillende plekke voorkom.

Ceri en Pelagatti [1] {p86} noem dit dat 'n groot gedeelte van die ontwerp van 'n verspreide databasis gedoen word deur gesentraliseerde databasisontwerpsmetodieke te gebruik. Die teenargument lei dat 'n spesiale benadering nodig is. Ons is duidelik besig met die ontwerp van iets ander as 'n gesentraliseerde databasis, hoekom dan die verspreidheidsfaktor ignoreer vir 'n groot gedeelte van die ontwerpproses? Ter ondersteuning hiervan is fase II se ontstaan te danke aan die uniekheid van 'n verspreide databasis.

Die uitset van fase II is 'n dokument wat die verspreiding van die elemente betrokke by die analise, nl. die organigram, funksies en entiteite sal toon, in terme van plekke, matrikse en ook in terme van numeriese affiniteite teenoor mekaar.

Die eerste gedeelte van hierdie fase is die ontleding van die geografiese plekke waar die verskillende organisasie-eenhede en elemente funksioneer en voorkom (plekontleding).

3.2 PLEKONTLEDING VIR VERSPREIDE DATABASISONTWERP

Martin [2] meen dat ons 'n matriksvoorstelling van plekke van voorkoms, teenoor organisasie-eenhede of afdelings moet opstel. Soos vooraf genoem, kan die organisasie-eenhede geïdentifiseer word deur na die organisasie-struktuur of organigram te kyk. Dit sal die afdelings, poste, mense en plekke identifiseer. Dit moet in gedagte gehou word dat funksies en entiteite reeds teenoor organisasie-eenhede geïdentifiseer was in die data-analise gedeelte in fase I. Deur dit te gebruik, kan ons 'n matriks van funksies en entiteite teenoor die plekke van voorkoms opstel. Nie alle entiteite of funksies waaruit 'n organisasie-eenheid bestaan, kom by alle plekke in die organisasie-struktuur voor nie. Ons kan bv. vind dat 'n personeel-keuringsfunksie nie by 'n tak, bv. Bloemfontein se personeel departement voorkom nie, alhoewel dit voorkom by ander personeel departemente in ander stede. 'n Personeel departement in bv. Johannesburg kan die personeel keuringsfunksie uitvoer.

Plekontleding se doel is 'n formele voorstelling van verspreiding. In die vorige fase het ons nie veel meer gedoen as die identifisering van plekke, departemente, afdelings, funksies en entiteite nie. Hierdie elemente moet nou formeel vertoon word, sodat die verwantskappe tussen die elemente teenoor plekke te voorskyn kom.

Hiervoor gaan ons weereens matriksvoorstelling toepas. Die aanname word gemaak dat die ontwerper sy eie keuse maak vir besigheids-terminologie, om die organisasie mee uit te beeld, bv. departemente kan bestaan uit afdelings of afdelings kan uit takke bestaan. Dit moet beskrywend wees sodat die verskillende organisasie-eenhede wat by die verskillende plekke gesetel is, duidelik voorgestel kan word. Hierdie beskrywing moet ook inval by die interne terminologie van die organisasie, sodat die bestuur en personeel ook die ontwerp en voorstelling kan verstaan.

Die volgende diagram toon 'n tipiese plekontledingsmatriks, wat gebruik kan word om verspreiding binne die organisasie uit te beeld. Let op dat die elemente van die matriks uit fase I se uitsette verkry word, nl. entiteitanalise en funksionele-analise. Ons vind ook dat organisasie-eenhede oor verskillende plekke versprei kan word, en dat dieselfde funksie of entiteit by verskillende plekke kan voorkom. Die plekontledingsmatriks lyk soos volg:

	PLEK 1	PLEK 2	PLEK n
ORGANISASIE EENHEID 1	X		
ORGANISASIE EENHEID 2		X	X
ORGANISASIE EENHEID n	X		
Funksie 1	X		X
Funksie 2		X	
Funksie n	X	X	
Entiteit 1	X		X
Entiteit 2			X
Entiteit n	X	X	

Ons kan ook ander inligting soos poste en persone in die plekontledingsmatriks voorstel. Enige bykomende inligting wat nodig of ontleed moet word, kan derhalwe bygevoeg word. Die matriks kan maklik geoutomatiseer word en met 'n rekenaarprogram geïmplementeer word in terme van 'n berekeningstabel (spreadsheet).

Nadat plekontleding gedoen is, moet ons oorgaan na 'n analitiese metode om die verspreiding van elemente te ondersoek. Dit word affiniteitsanalise genoem. Dit verskaf 'n numeriese maatstaf van die geneigdheid van elemente om bymekaar geplaas te word, of bymekaar te hoort a.g.v. die eienskappe van die elemente.

3.3 AFFINITEITSANALISE

Martin [2] (p314) noem dat affiniteitsanalise gebruik kan word vir die analise en die saamgroepering van elemente of objekte. Hierdie saamgroepering kan gebruik word om die fisiese verspreiding van die elemente voor te stel. Hy noem spesifiek dat hierdie analise 'n aanknopingspunt kan wees vir gedetailleerde datamodellering. Affiniteitsanalise pas dus uitstekend in hierdie fase van die ontwerpmetodologie, (fase II) omdat die volgende fases konsentreer op datamodellering, fragmentasie en toewysings, en van verspreidingsanalise gebruik maak.

Alhoewel Marin se affiniteitsanalise, entiteite teenoor die aktiwiteite wat hulle gebruik voorstel, gaan ons nie aktiwiteite as een van die faktore gebruik nie, maar wel plekke van voorkoms. Sodoende kan ons verspreidingsanalise van entiteite teenoor plek, en funksies teenoor plek uitbeeld. Die laaste stap kan wel ook entiteite teenoor funksies uitbeeld om 'n meer volledige beeld van affiniteite tussen objekte te verkry.

As voorbeeld kyk ons na twee entiteite E1 en E2 wat by verskillende plekke in die organisasiestruktuur voorkom. As hierdie entiteite nooit saam voorkom by die selfde plek nie, is hulle affiniteit gelyk aan nul (0). As E1 en E2 altyd by dieselfde plekke voorkom, is hulle affiniteit gelyk aan een (1).

Die kompleksiteit van affiniteitanalise word verhoog as E1 en E2 by verskillende plekke voorkom. Verskillende entiteite se affiniteit teenoor mekaar kan ontleed word deur 'n formule te gebruik wat die plekke van voorkoms in ag neem:

$$\text{Affiniteit van E1 tot E2} = \frac{a(E1, E2)}{a(E1)}$$

Waar:

$a(E1)$ = aantal plekke waar E1 voorkom.

$a(E1, E2)$ = aantal plekke waar beide E1 en E2 voorkom.

As E1 by Johannesburg, Durban en Kaapstad voorkom en E2 by Johannesburg, Kaapstad, Durban en Bloemfontein, is die affiniteit van E1 tot E2 = 1. Dit word soos volg bereken:

$$\text{Affiniteit van E1 tot E2} = 3 / 3 = 1.0.$$

In bg. voorbeeld verskil die affiniteit van E2 tot E1 van die affiniteit van E1 tot E2 : Affiniteit van E2 tot E1 = $3 / 4 = 0.75$. Hieruit sien ons dat E2 nie by al die plekke voorkom waar E1 voorkom nie. Affiniteitsanalise is maklik om te outomatiseer omdat die berekeninge met 'n rekenaarprogram gedoen kan word. Dit sal moeilik vir die ontwerper wees om duisende entiteite se affiniteite teenoor mekaar, met die hand uit te werk. Hierdie gedeelte van die metodologie behoort dus geoutomatiseer te word.

Deur hierdie analise op alle objekte soos entiteite en funksies toe te pas, kan ons die objekte wat saamhoort bymekaar groepeer. Objekte met hoë affiniteite behoort bymekaar groepeer te word, en objekte met 'n lae affiniteit moet geskei word. 'n Afsnypunt vir groepering behoort gevind te word, d.w.s by watter affiniteitsfaktor moet ons die objekte skei of saam groepeer. Dit kan bv. gebaseer word op die drakrag van die kleinste rekenaar in die netwerk.

Ons kan die affiniteitsanalise verder in die metodologie gebruik as hulpmiddel vir replisering en fragmentering van die databasis. Affiniteitsanalise is nie 'n volkome maatstaf vir saamgroepering nie, maar word hoofsaaklik gebruik om 'n goeie empiriese aanduiding te gee vir groepering. Ander faktore soos oorbodigheid, kan veroorsaak dat gedeeltes van ons affiniteitsanalise se resultate, minder belangrik word.

'n Verdere analise kan uitgevoer word deur die affiniteite tussen objekte weereens met wiskundige metodes te manipuleer, en 'n totale beeld van saamgroepering op te stel. Ons noem hierdie analise "geweegde-affiniteitsanalise" of saamgroepering gebaseer op affiniteit.

3.3.1 SAAMGROEPERINGSALGORITME GEBASEER OP AFFINITEIT

Martin [2] [p315] noem dat ons 'n saamgroeperingsalgoritme gebruik kan word om die fisiese groepering van objekte te kan ontleed. Hiervoor word affiniteite tussen objekte gebruik. Die objekte met die hoogste affiniteite vorm die kern van die verspreidingsgroepeerings waarom ander entiteite gegropeer word.

Die voorgestelde saamgroeperingsalgoritme funksioneer soos volg (Martin [2] [p315]):

1. Sorteër affiniteite in terme van hulle affiniteitsfaktore van hoog na laag, d.w.s van affiniteitsfaktor 1 tot affiniteitsfaktor 0.
2. Identifiseer die affiniteite waar een van die objekte van die paar, reeds vooraf voorkom in die gesorteerde lys van affiniteite.
3. Toets of die ander objek verbind aan die objek wat reeds voorkom, liewers met die vorige paar of die huidige paar gegropeer moet word. Hiervoor gebruik ons geweegde affiniteitsanalise tussen die vorige paar en die objek (in volgende voorbeeld verduidelik).
4. As die geweegde affiniteit hoër is as enige ander affiniteite tussen ander pare wat die een objek bevat wat reeds voorkom, moet die ander objek by

die vorige voorkoms gegroepeer word, anders moet die objek in die oorspronklike saamgroepering bly.

'n Voorbeeld van gesorteerde affiniteitspare wat bestaan uit entiteite wat by verskillende plekke in 'n organisasie voorkom is:

ENTITEIT PARE	AFFINITEITSFAKTOR
E1, E4	0.92
E11, E8	0.90
E6, E7	0.88
E10, E12	0.87
E2, E8	0.85
E7, E4	0.76
E8, E10	0.74

'n Geweegde affiniteitsanalise op paar E2, E8 kan gedoen word, om uit te vind waar die paar gegroepeer moet word:

$$\frac{a(E2, E11) \times a(E11) + a(E2, E8) \times a(E8)}{a(E11) + a(E8)}$$

$$= \frac{(0.34 \times 3) + (0.85 \times 48)}{3 + 48}$$

$$= 0.82$$

Waar: a(E, E) die affiniteit tussen die paar entiteite is. a(E1) die aantal plekke is waar die enkele entiteit E voorkom.

$$a(E2, E11) = 0.34$$

$$a(E11) = 3 \text{ (E11 word by 3 plekke gebruik)}$$

$$a(E2, E8) = 0.85$$

$$a(E8) = 48 \text{ (E8 word by 48 plekke gebruik)}$$

Gebaseer op die antwoord van 0.82 (Hoë affiniteit) wat laer is as $a(E2,E8)$ maar hoër as $a(E8,E10)$, moet E2 saam met die paar $a(E11,E8)$ gegropeer word om 'n groep E2,E11,E8 te vorm.

Die resultaat van geweegte-affiniteitsanalise is groepies objekte wat by fisiese plekke gegropeer word, om die mees effektiewe verspreiding van die objekte te reflekteer. Alternatiewe groeperings kan ook hiermee ontleed word. Martin [2] (p314) noem dat geweegte-affiniteitsanalise goeie resultate in die praktyk gelewer het. Dit behoort dus ook bruikbaar te wees binne die konteks van hierdie metodologie.

3.4 OPSOMMING

Hierdie hoofstuk behandel die tweede fase van die ontwerpmetodiek nl. verspreidingsanalise. Hiervoor het ons die inligting wat ons in fase I bekom het, gebruik as inset vir fase 2. Die verspreiding-veranderlike is op die voorgrond geplaas, deur plekanalise op die elemente binne die organisasie uit te voer. Daarna het ons affiniteitsanalise gedoen om 'n numeriese voorstelling van die elemente se saampasbaarheid of affiniteit, te verkry. Ons het dus uitgevind watter elemente saamhoort en watter nie bymekaar hoort nie.

Deur 'n algoritme op affiniteitsanalise toe te pas, was geweegde-affiniteitsanalise gedoen wat die groepering van elemente om hoë-affiniteit kerns aantoon. Die uitsette van fase II kan tipies die volgende insluit:

- 1) 'n Plekontledingsmatriks.
- 2) Affiniteitsanalise van entiteite en funksies.
- 3) Saamgroeperingsanalise van entiteite en funksies.

Hierdie uitsette word in latere fases gebruik om spesifiek die verspreiding van die elemente aan te toon. Deur fase II te voltooi, verseker ons dat ons alle faktore betreffende die verspreiding van die organisasie verstaan. Dit kan beskou word as analise en kennis-insameling van verspreiding, om sodoende meer insig in die struktuur van die verspreide databasis te verkry.

HOOFSTUK 4

FASE III : KONSEPTUELE-LOGIESE ENTITEITVERWANTSKAPMODELLERING

Opsomming

Ons bespreek die implementering van die terme "logiese model" en "konseptuele model". Daarna word entiteitverwantskapmodelle (EV-modelle) ontleed, om 'n konseptuele-logiese datamodel vir 'n verspreide organisasie te modelleer. Die keuse vir die gebruik van die EV-model word ook geregverdig. Stappe vir die opstel van die konseptuele-logiese model word ook bespreek. Die resultaat van hierdie fase is 'n uitgebreide EV-model wat 'n konseptuele-logiese beeld van verspreide organisatoriese data voorstel. Ons kyk ook na die moontlikheid van transformasie vanaf die EV-model na die relasiemodel, d.w.s. vanaf 'n konseptuele-logiese na 'n fisies implementeerbare model. Hierdie fase se uitset is konseptuele-logiese EV-modelle met die moontlikheid van transformasie na logies-fisiese relasiemodelle.

4.1 INLEIDING

Dit moet in gedagte gehou word dat die EV-model gebruiker georiënteerd is, en nie vir programmeerders ontwikkel is nie. Hierdie hoofstuk verskaf 'n bespreking van die gebruik van 'n uitgebreide EV-model om die konseptuele-logiese struktuur van 'n verspreide databasis te modelleer. Ons ontleed ook die term "konseptuele-logiese model" omdat dit van toepassing is in hierdie hoofstuk. 'n Uitgebreide EV-model is deur Teorey, Yang en Fry [31] (p197) voorgestel, wat geskik behoort te wees vir die modellering van verspreide databasisse. Hulle beskryf ook 'n metodiek waardeur die EV-model opgestel kan word, sodat dit later na 'n relasiemodel getransformeer kan word.

Ons gebruik die resultate van fase I en II (uitsette te doen met entiteitanalise) as inset vir hierdie fase. In hierdie fase (fase III) gaan ons 'n konseptuele-logiese EV-model bou, wat die verspreide datastruktuur van die organisasie sal reflekteer, deur om die vooraf gedefinieerde entiteite as boublokke te gebruik.

Dit is belangrik om 'n konseptuele model soos die EV-model te kan transformeer na 'n fisiese implementeerbare model soos die relasiemodel, alhoewel die relasiemodel ook 'n logiese model is.

Daar is reeds heelwat navorsing gedoen oor transformasie tussen modelle. Ons gaan vlugtig na hierdie aspek kyk, en transformasie vanaf die EV-model na die relasiemodel bespreek. Die uitset van hierdie fase is 'n konseptuele-logiese uitgebreide EV-model van 'n verspreide organisasie se datastruktuur, wat dan omgeskakel kan word na 'n logiese-fisiese relasiemodel.

Om terminologie te standardiseer, moet ons kyk na wat bedoel word met die term : "konseptuele-logiese model". In die volgende gedeelte vind 'n bespreking van die term "konseptuele-logiese model" plaas.

4.2 'n KONSEPTUELE-LOGIESE MODEL VIR VERSPREIDE DATABASISSE

Die term "konseptuele-logiese model" is 'n kombinasie van modellering met die begrippe "logiese" en "konseptuele". Die term "konseptueel" spruit uit die konseptualiserings-vermoëns van die mens. Dit behels 'n vlak van abstrakte denke, waarby identifisering en meegaande verwantskappe, tussen die elemente beskryf word. Dit behels ook 'n breër vlak van abstraksie as wat in die realistiese werklikheid, (fisiese wêreld) betrokke is (Korth en Silberschatz [39] (p4)).

Die konsep van 'n "logiese model" toon 'n teenstelling met die konsep van 'n "fisiese model". Dit het dus niks te doen met die fisiese implementering van 'n databasis nie, maar wel met die logiese beeld van die databasis. Hierdie logiese model is 'n voorstelling van die werklikheid. Soos vooraf bespreek, (hoofstuk 1 - 1.5.4.1 (p27)) kan ons logiese modelle verdeel in 1) objek-gebaseerde modelle en 2) rekord-gebaseerde modelle. Onder kategorie 1 vind ons die EV-model en onder kategorie 2 vind ons die relasiemodel (Korth & Silberschatz [39] (p6-8)).

Daar is ook 'n verskil tussen die modelle in terme van konseptueel vs. fisies, in die lig daarvan dat die EV-model as konseptuele denkbeeldmodel gebruik word, en die relasiemodel as fisiese implementeerbare model gebruik word.

Vervolgens gaan ons die gebruik van die EV-model in die skeep van konseptuele modelle of skemas ontleed. Hierdeur kan ons dan toon dat dit wel by die tradisionele beskouing van databasisontwerp nl. skema /subskema vs. modelle, inpas.

4.2.1 DIE EV-BENADERING EN DIE OPSTEL VAN KONSEPTUELE SKEMAS VIR VERSPREIDE DATABASISSE

Die klem in hierdie afdeling rus hoofsaaklik op die term "konseptueel", in beide modellering en skema gerigte benaderings vir die ontwerp van verspreide databasisse.

Ons kan aantoon dat die EV-benadering wel gebruik kan word vir die opstel van konseptuele skemas. Dit behoort ook van toepassing te wees by die opstel van konseptuele skemas vir verspreide databasisse. Die term "konseptuele skema" kan omskryf word as : die formele datamodel wat in die gebruikers en ontwerpers se gedagtes gevorm is. Martin [2] noem dat hierdie model 'n model van die werking van die organisasie moet wees.

Dit word dus hoofsaaklik gebruik as 'n hulpmiddel vir die ontwerp en implementering van die databasis, sodat die databasis-konsepte betrokke, gehanteer kan word. As verdere illustrasie van die nut van die EV-model in die ontwerp van die konseptuele skema, kan ons kyk na Sakai, Kondo en Kawasaki [29] (411) se CSDA. Hulle het die CSDA (Conceptual Schema Design Aid) ontwikkel wat die EV-model gebruik vir die ontwerp van die konseptuele skema van 'n databasis. Hiervoor gebruik hulle 'n uitgebreide EV-model.

Die CSDA stelsel word gesentreer om die CSD, oftewel die Conceptual Schema Dictionary. Verder gebruik hulle 'n interaktiewe skemadefinisietaal sodat die gebruiker maklik toegang tot die verskillende ontwerpsprosesse het. Hierdie stelsel behoort van toepassing op verspreide databasisse te wees, deur die definisie uit te brei om die "plek"-veranderlike te bevat.

Atzeni, Batini, Lenzerini en Villanelli [28] (p375) verskaf ook 'n stelsel wat gebruik kan word vir die konseptuele ontwerp van 'n databasis. Hulle stelsel word INCOD genoem, oftewel "Interactive Conceptual Design" waarin hulle ook 'n uitgebreide EV-model gebruik. 'n Bykomende dimensie in konseptuele ontwerp word verskaf deur albei van bg. implementasies, in die sin dat albei transaksie-analise, doen d.w.s konseptuele transaksie-analise. Beide INCOD en CSDA hanteer ook transaksie-modellering. Hierdie aspek van konseptuele transaksie-ontwerp sal later meer volledig bespreek word, omdat dit 'n spesifieke rol in verspreide databasisontwerp speel (by fragmentering en plasing).

Deur bg. implementasies as voorbeeld te neem, kan ons regverdiging aanvoer, dat die EV-model wel gebruik kan word vir die implementering van konseptuele skemas of modelle vir verspreide databasisse. Hulle sal net aangepas moet word om verspreidheid te reflekteer.

As verdere regverdiging gaan ons volgende kyk na redes hoekom die EV-model gekies behoort te word, om die rol van 'n konseptuele logiese model binne verspreide databasisontwerp te vervul. Dit sal uitgebeeld word in terme van die voordele wat die EV-model inhou.

4.3 VOORDELE VAN DIE EV-MODEL VIR VERSPREIDE DATABASISONTWERP

Om die rol van 'n konseptuele-logiese model te vervul, kan 'n hele paar voorstellingsmodelle gebruik word, wat in die verlede voorgestel is. 'n Literatuur analise sal aantoon dat Chen [22] se EV-model heelwat voorkom.

Chen [19] noem in die voorwoord van sy boek "Entity-relational approach to information modeling and analysis", dat die grootste probleem in databasisontwerp, die analise en modellering van die datamodel is. Die rede hiervoor is die kommunikasiegaping tussen die gebruiker en die stelselontleder of databasis ontwerper. Daar word dus gesoek na 'n voorstellingshulpmiddel wat die gebruiker se inligtingsbehoefte in eenvoudige maar kragtige terme kan omskryf. Die metode moet ook effektief integreerbaar met 'n bestaande databasismodel, soos die relasiemodel, wees. Martin [2] (p 279) definieer die gebruik van die entiteits-verwantskapsmodel soos volg:

"Entity-Relationship Modeling creates a chart of the entities and their relationships which is an overview of the data that must be stored in the enterprise data bases. The entities are associated with business functions in a matrix and this matrix is clustered to find naturally cohesive groups of entities and functions."

Martin [2] gebruik die EV-model dus binne 'n globale konteks om die volledige struktuur van 'n organisasie uit te beeld.

Hy konsentreer nie net op die datamodelleringsienskappe nie, maar gebruik dit as 'n voorstellingsmodel vir die totale organisasie. Dit kan ook gebruik word om die totale verspreide struktuur van die organisasie te modelleer.

Teichroew, Germano en Luca [20] beskryf verskeie toepassings waarvoor die EV-model as basis gebruik is, vir dokumentasie en analise van dataverwerkingstelsels. Hulle bespreek die feit dat die EV-benadering op bestaande databasismodelle soos die CODASYL-DBBS geïmplementeer kan word, en 'n meer algemene en natuurlike beeld van 'n dataverwerkingstelsel verskaf. Dit behoort ook van toepassing op verspreide organisasies te wees.

Prakties, in terme van verspreide databasisse, is EV-gerigte modelle ook reeds gebruik as voorstellingsmodel vir verspreide databasisse bv. Elmasri, Cog en Rahini [26] het die entiteit-kategorie-verwantskapmodel (veralgemening van die EV-model) gebruik as voorstellingsmodel op Honeywell se DDTS-stelsel (Distributed Database Testbed System).

Die ontstaan van die EV-model het 'n nuwe dimensie in databasisontwerp verskaf, omdat dit as 'n suiwer konseptuele model gebruik kan word. Alhoewel dit 'n ekstra stap in 'n ontwerpmetodologie meebring (bv. konseptuele-logiese EV-modellering i.p.v. die opstel van die logiese skema), kan dit geregverdig word, deur die voordele wat dit tot gevolg het. Tradisioneel was 'n fisies-implementeerbare model soos die relasiemodel gebruik om konseptuele modellering te doen. Die gebruik van 'n suiwer konseptuele model bring nie werklik 'n ekstra stap in 'n ontwerpmetodologie nie, omdat ons direk na 'n fisiese implementeerbare model kan transformeer soos later getoon sal word.

Teorey, Yang en Fry [31] (p198) noem dat een van die grootste voordele wat die EV-model vir die databasis ontwerper inhou, die feit is dat 'n bo-na-onder ontwerpbenadering (Afdeling - 1.5.5) daarmee gevolg kan word. Die bruikbare aspek van stapsgewyse verfyning en abstraksie kom dus ter sprake.

Hulle noem ook dat die EV-model baie suksesvol is om die kommunikasie tussen die ontwerper en die gebruiker te maksimaliseer. Die model is dus gebruiker georiënteerd. Dit het ook 'n eenvoudige diagrammatiese voorstellings-struktuur.

Die model is ontwerp om persepsies van die realistiese werklikheid (regte wêreld) te reflekteer, om sodoende 'n organisatoriese model of skema te vorm. 'n Skema in terme van 'n EV-model, word gedefinieer as : 'n logiese en konseptuele beskrywing van data, wat definisies van entiteite, verwantskappe en ander logiese strukture bevat (Aangepas uit Cardenas [4] (p510)).

Ons vind dus dat die EV-model uiters geskik is vir die konseptuele ontwerp van 'n verspreide databasis a.g.v. hierdie redes:

1. Die basiese EV-model kan maklik gepermuteer word om spesifieke behoeftes en omstandighede van ontwerpaksies te dek.
2. Die EV-model se struktuur is eenvoudig en kan maklik deur die gebruikers en ontwerpers van 'n databasis verstaan en gebruik word.
3. Die EV-model kan gebruik word as tussenganger- of oorskakelingsmodel tussen verskeie ander databasismodelle nl. tussen netwerk-, hiërargiese- en relasie-modelle. Dit stel ons in staat om verskillende modelle te kombineer, wat veral nuttig kan wees in heterogene verspreide databasisse.
4. Die EV-model kan geoutomatiseer word (bv. CSDA - 4.2.1). Ons kan dus 'n gerekenariseerde ontwerpshulpmiddel ontwikkel wat die behoeftes van die ontwerper kan bevredig. Ons behoort selfs sover te kan gaan as om 'n toepassingsgenerator met kunsmatige intelligensie in die stelsel in te bou.

5. Die EV-model kan binne bestaande standarde gebruik word, en kan selfs ontwikkel as 'n globale standaard vir konseptuele databasisontwerp.
6. Fisiese verspreiding kan uiters eenvoudig in die EV-model voorgestel word, deur entiteite te gebruik en die model te fragmenteer vir verskillende plekke.
7. Die EV-benadering bied 'n totale en geïntegreerde benadering vir databasisontwerp wat onafhanklik van die fisiese databasismodel is.

Ons sien dus dat die gebruik van die EV-model 'n baie bruikbare konseptuele benadering is, wat die ontleding van 'n verspreide organisasie se struktuur of 'n werklikheidsprobleme kan vergemaklik. Vir implementasie van die EV-benadering, moet ons die ontwerpsbenadering spesifiseer. Dit word volgende beskryf in terme van die ontwerpsbenadering wat vir die EV-model, gevolg word.

4.4 VERSPREIDE DATABASIS GERIGTE EV ONTWERPSBENADERING

Alhoewel die bo-na-onder, onder-na-bo en gekombineerde benaderings tot minder en meerdere mate van sukses gebruik is om 'n databasis te ontwikkel, was die onder-na-bo benadering meer gewild (Teorey, Yang en Fry [31] [p197]). Dit spruit waarskynlik uit die gebruik van datawoordeboeke waarin laëvlak elemente van 'n databasis vasgelê is, en stelselmatig verfyn is om 'n bruikbare databasis-skema te vorm.

Tradisionele ontwikkelingsbenaderings t.o.v. bv. relasie-databasisse, het die data vanaf die laagste vlak (data-elemente) saamgevat, om onder-na-bo benaderde relasies te vorm. Een van die probleme van verspreide databasisse, is die aantal en kompleksiteit van data-elemente op die laer ontwerpsvlakke.

Dit is moeilik en amper menslik onmoontlik om 'n groot aantal data-elemente van onder-na-bo te probeer saamvat, a.g.v. die resulterende kompleksiteit van die relasies. Teorey, Yang en Fry (31) (p198) ondersteun hierdie beginsel deur te noem:

" Although the tradisional process is vital to the design of relational databases, its complexity, particularly in large databases, can be overwhelming to the point where practical designers often do not bother to master it or even use it with any regularity."

Hulle noem dit dat die ontstaan van 'n konseptuele model soos die EV-model, gelei het na die moontlikheid om 'n gekombineerde benadering in databasisontwerp te volg (bv. bo-na-onder en onder-na-bo deur die kombinerings van die EV-model en relasiemodel). Al die voordele van stapsgewyse verfyning, data abstraksie, en bo-na-onder samevatting is dus nog ter sprake.

Die EV-model gebruik hoofsaaklik 'n bo-na-onder benadering wat op redelike hoëvlak entiteite (abstraksie) gebaseer word. Die analise van die stelsel is dus minder kompleks, omdat ons stapsgewys vereenvoudig en nie wegspring met 'n groot aantal laëvlak data-items nie. Ons vereenvoudig dus op 'n tradisionele bo-na-onder wyse waar stapsgewyse verfyning ter sprake kom.

Ons kyk vervolgens na die EV-model se basiese struktuur sodat ons 'n beeld van die EV-model as bruikbare hulpmiddel vir verspreide databasisontwerp kan vorm.

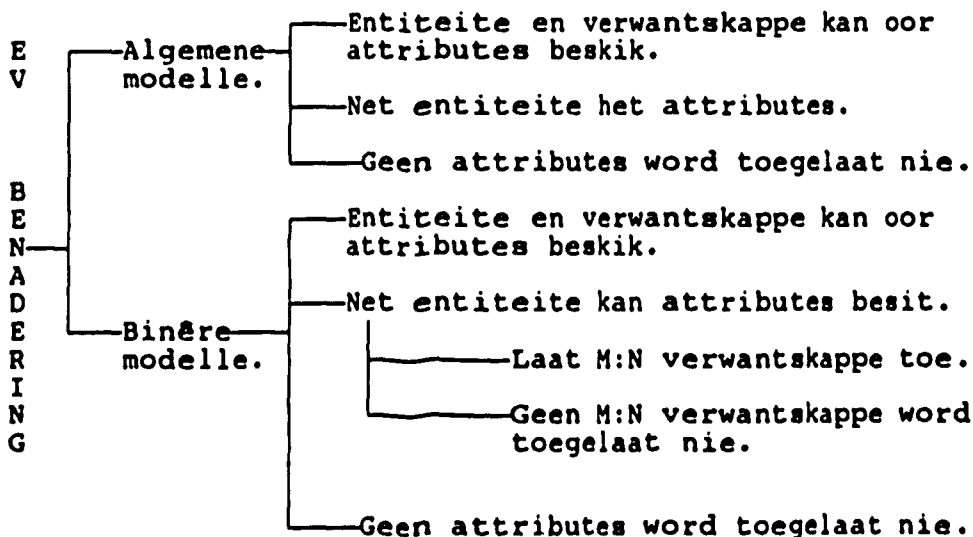
4.5 DIE ENTITEITVERWANTSKAPMODEL

As die entiteitverwantskapmodel ontleed word, behoort ons uit te vind binne watter raamwerk die EV-model pas, in terme van die moontlikhede vir data voorstelling. Verskeie persone het

die basiese EV-model as boustene vir hulle eie konseptuele modelle gebruik.

Die beste voorbeeld hiervan kan gevind word in James Martin se "Information Engineering"-strategie [2] waarin hy die EV-model gebruik om die hele inligting-struktuur van 'n organisasie uit beeld. Chen [21] (p19) meen dat verskillende definisies aan die terme "verwantskap", "entiteit" en "attribuut" geheg word. Om verwarring te voorkom, het hy 'n standaard raamwerk geskep, wat gebruik kan word om enige EV gebaseerde model binne 'n raamwerk te plaas. Hy klassifiseer die bruikbare EV-benaderings binne 'n gestandaardiseerde raamwerk wat later bespreek sal word. Sy raamwerk word gebaseer op reëls wat toegepas kan word op die model se verwantskappe en attributes.

Dit is wenslik om 'n eiesoortige EV-model te definieer, wat op 'n verspreide databasis gebruik kan word, en wat binne Chen [21] se raamwerk sal pas. As so 'n model nie beskikbaar is nie, of ontoepaslik is, moet ons 'n spesifieke model kan ontwikkel, of 'n bestaande model kan aanpas om verspreidheid te reflekteer. Dit word in die volgende gedeeltes van hierdie hoofstuk gedoen. Die raamwerk wat deur Chen [21] voorgestel word, (vereenvoudig) waarin EV-modelle geplaas kan word, lyk soos volg:



Chen se oorspronklike EV-model [22] word in die kategorie van 'n algemene model geplaas. 'n Algemene model is 'n model waarin 'n verwantskap tussen meer as twee entiteite voorgestel kan word, terwyl 'n binêre model 'n maksimum van twee entiteite in 'n verwantskap toelaat. Hierdie raamwerk toon dat 'n model wat die basiese definisies van entiteit, verwantskap en attribuut hanteer, verskillende implementasiereëls het, en kan sodoende aangepas word om verskillende toestande en situasies te bevredig. Daar is dus nie 'n "beste" EV-model nie maar net meer toepaslike modelle. 'n Verspreide databasis EV-model moet oor sekere eienskappe beskik wat die verspreidheidsdimensie sal kan hanteer.

Chen [22] se oorspronklike EV-model dien as basis vir die uitgebreide model van Teorey, Yang en Fry [31]. Die basiese EV-model bestaan uit entiteit-stelle, entiteite, attributes, verwantskappe, verwantskap-stelle en kardinaliteite. Hierdie elemente kan grafies binne 'n diagram voorgestel word (Hoofstuk 1 : figuur 1 - 1.5.6.1). Ons gaan nou die basiese model verder bespreek.

4.5.1 ENTITEITE, ATTRIBUTES EN VERWANTSKAPPE

Chen [22] se EV-model het entiteite, attributes en verwantskappe as boublokke gebruik. Verder is 'n versameling van dieselfde entiteite of verwantskappe in terme van entiteit-stelle of verwantskap-stelle geklassifiseer. Ons vind later dat entiteit-stelle wegval en ingesluit kan word binne die definisie van 'n entiteit.

Korth en Silberschatz [39] (p21-24) het die volgende definisies aan bg. terme: 'n Entiteit is 'n objek wat bestaan, en verskil van ander objekte, hetsy die objekte konkreet of abstrak is. 'n Entiteit-stel is 'n

versameling of groepering van dieselfde entiteite. 'n Entiteit kan opgedeel of voorgestel word, d.m.v. attributes. 'n Attribuut van 'n entiteit kan gedefinieer word in 'n domein van moontlike waardes. 'n Verwantskap is 'n assosiasie tussen entiteite. Dit beskryf dus 'n koppeling of afhanklikheid tussen entiteite.

Daar moet ook verstaan word dat 'n entiteit 'n verwantskap met homself kan hê, bv. entiteit "WERKNEMER" (bv. 'n voorman) kan 'n verwantskap bevat wat toon dat 'n ander werknemer (bv. operateur) wat met dieselfde entiteit beskryf word, vir die voorman werk.

4.5.2 KARDINALITEITE

Verder beskryf die EV-model kardinaliteite tussen entiteite d.m.v. die verbindingsverwantskap tussen entiteite. As ons twee entiteit-stelle A en B beskou, vind ons die volgende moontlike kardinaliteite (Korth en Silberschatz [39] {p25-26}) :

Een tot Een: Een entiteit uit entiteit-stel A kan verbind word met net een entiteit van entiteitstel B en andersom.

Een tot Meer: Een entiteit van entiteit-stel A kan verbind word met enige hoeveelheid entiteite in entiteit-stel B. Een entiteit van entiteit-stel B kan met net een entiteit van entiteitstel A verbind.

Meer tot Een : Een entiteit van entiteit-stel A kan met hoogstens een entiteit van entiteit-stel B geassosieer word terwyl een entiteit van entiteit-stel B met enige hoeveelheid

entiteite van entiteit-stel A kan verbind.

Meer tot Meer : Enige entiteit uit entiteit-stel A kan geassosieer word met enige hoeveelheid entiteite uit entiteit-stel B, en andersom.

Daar moet onthou word dat die voorstelling van kardinaliteite tussen entiteite altyd die realistiese werklikheid (regte wêreld) moet voorstel.

4.5.4 SWAK ENTITEITE

'n Swak Entiteit kan beskryf word as: 'n entiteit wat in verwantskap met 'n ander entiteit verkeer, sodat die swak entiteit slegs kan bestaan as die ander entiteit bestaan. Die sterker entiteit is dus die "eienaar" van die swak entiteit. Korth en Silberschatz [39] (p27) gebruik die terme dominante entiteit en ondergeskikte entiteit. Die ondergeskikte entiteit se bestaansreg is dus gesetel in die feit dat die dominante entiteit bestaan. As die dominante entiteit wegval, moet die ondergeskikte entiteit ook wegval.

Ons het gekyk na die basiese elemente van die EV-model, en gaan nou kyk na 'n uitgebreide model wat meer doelgerig kan funksioneer, en gebruik kan word vir die ontwerp van verspreide databasisse.

4.6 'n UITGEBREIDE ENTITEITVERWANTSKAPMODEL VIR VERSPREIDE DATABASISSE

Ons beskou die uitgebreide EV-model wat deur Teorey, Yang en Fry [31] (p198) voorgestel is. Hulle is van mening dat 'n

uitgebreide model nodig is om 'n effektiewe konseptuele skema voor te stel, wat die regte wêreld sal kan modelleer. Hulle beskryf die gebruik van die tradisionele EV-model soos volg:

" Using the ER model as a conceptual schema representation, however, has proved difficult because of the inadequacy of the modeling constructs. "

Die Uitgebreide EV-model (UEV-model) bied spesiale semantiese strukture waarmee opsionele verwantskappe, drieledige (ternary) verwantskappe en subtypes of subkategorië uitgebeeld kan word. Net soos die tradisionele EV-benadering wat vooraf beskryf is, gebruik die uitgebreide model ook entiteite en verwantskappe as basiese uitgangspunt, vir die opstel van die model.

Chen [22] se oorspronklike EV-model is ook steeds geldig. Die uitgebreide model bevat die volgende konsepte nl. entiteite, verwantskappe, attributes (beskrywing en identifiseerders), objek-klasse, verwantskapsgraad, verbintenisgraad en lidmaatskapsklasse. 'n Diagrammatiese voorstelling van bg. konsepte, word in die artikel [31] getoon. Ons kyk nou verder na elk van hierdie konsepte wat deur Teorey, Yang en Fry [31] beskryf is.

4.6.1 ENTITEITE, ATTRIBUTES EN VERWANTSKAPPE

Entiteite is enige persoon, plek, ding of gebeurtenis wat belangrike inligting bevat. Attributes word gebruik om die entiteite verder te beskryf d.m.v. beskrywende naamwoorde soos : naam, plek, kleur, gewig, nommer, ens.

Ons sien dus dat entiteite beskryf kan word deur gebruik te maak van 'n aantal attributes. Teorey, Yang en Fry [31] (p200) identifiseer twee tipes

attributes nl. 1) identifiseerder-attributes en 2) beskrywer-attributes. Identifiseerder-attributes word gebruik om die voorkoms van verskillende entiteite van dieselfde tipe te onderskei. Beskrywer-attributes word gebruik om die bestaan van entiteite in meer besonderheid te beskryf.

Deur na die waardes van entiteite se identifiseerder-attributes te kyk, kan ons "sterk" of "swak" entiteite identifiseer. 'n Sterk entiteit word gekenmerk deur interne identifiseerders wat die voorkoms van 'n entiteit uniek identifiseer. Die voorkoms of bestaan van 'n swak entiteit is afhanklik van die identifiseerder-attributes van 'n ander sterk entiteit.

Verwantskappe beskryf werklike verbintenisse tussen entiteite in terme van semantiese stellings. As gevolg hiervan kan ons 'n verbintenis tussen entiteite identifiseer, deur ook inligting i.v.m. die graad van verbintenis te toon. 'n Voorbeeld hiervan is : in sekere wêrelddele kan entiteite "MAN" en "VROU" verbind wees met verwantskap "GETROUD" waar die graad van verbintenis 1:N is, (een man mag met meer as een vrou getroud wees) terwyl die verbintenis-graad in die Westerse wêreld moet 1:1 wees (een man mag net met een vrou getroud wees).

4.6.2 OBJEK-KLASSE

Objek-klasse is 'n uitgebreide konstruk wat by Chen se oorspronklike EV-model gevoeg is, om skema-voorstelling te vergemaklik. Dit is oorspronklik deur Navathe en Cheng (40) voorgestel. Ons vind twee tipes objek-klasse nl. 1) Substel-hierargië en 2)

Veralgemening-hierargië.

Substel-hierargië word gevind waar die entiteit E1 'n substel van 'n ander entiteit E2 is sodat by elke voorkoms van entiteit E1, entiteit E2 ook altyd voorkom. Die voorkoms van 'n generiese entiteit impliseer dus die voorkoms van potensieële oorvleuelende entiteite bv. entiteit "WERKNEMER" kan 'n substel-hierargie met entiteit "STUDERENDE WERKNEMER" of "AFGETREDE WERKNEMER", vorm.

Veralgemening-hierargië word gevind waar entiteit E 'n veralgemening is van ander entiteite E1, E2, E3...E-n sodat elke voorkoms van entiteit E ook 'n voorkoms is van net een entiteit van E1 of E2 of E3.. of tot ..E-n. Dit kom voor wanneer 'n generiese entiteit verdeel word a.g.v. verskillende waardes van 'n oorvleuelende attribuut bv. die entiteit "WERKNEMER" is 'n veralgemening van entiteite "SEKRETAESSE", "TEGNIKUS", "REKENMEESTER" ens. Die voorkoms van entiteit "WERKNEMER" kan dus net die voorkoms van een entiteit "SEKRETAESSE" of een entiteit "TEGNIKUS" of een entiteit "REKENMEESTER" aantoon.

4.6.3 VERWANTSKAPSGRAAD

Die verwantskapsgraad toon die aantal entiteite aan, wat deel is van die verwantskap, d.w.s dit is aan die verwantskap gekoppel. As N aantal entiteite aan 'n verwantskap gekoppel is, is die verwantskapsgraad "N-LEDIG". Teorey, Yang, en Fry (31) (p210) gebruik net eenledig, tweeledig en drieledige verwantskapsgrade vir hulle UEV-model. 'n Eenledige verwantskap kom voor waar die entiteit 'n verwantskap met homself het.

4.6.4

VERBINTENISGRAAD

Die verbintenisgraad spesifiseer die aantal voorkomste van een entiteit, in teenstelling met die voorkoms van 'n gekoppelde entiteit, d.w.s die aantal voorkomste van entiteite wat kan voorkom met die bestaan van ander entiteite. Die waardes wat die verbindinggraad aantoon is of "EEN" of "BAIE". Die fisiese hoeveelheid wat die term "BAIE" beskryf, word die kardinaliteit van die verbinding genoem, wat in terme van 'n lae en 'n hoë grens uitgedruk kan word.

4.6.5

LIDMAATSKAPSKLASSE

Lidmaatskapsklas toon aan of die "EEN" of "BAIE" deel van die verwantskap opsioneel is of nie. 'n Entiteit aan die "EEN" kant van die verwantskap, wat altyd in die stelsel moet bestaan, toon 'n nie-opsionele lidmaatskapsklas. As die entiteit aan die "EEN" kant van die verwantskap nie in die stelsel hoef te bestaan nie, is dit opsioneel. Aan die "BAIE" kant van die verwantskap is die entiteit opsioneel as nie een van die moontlike, "BAIE" entiteite moet voorkom nie, bv. entiteit "KANTOOR" word met verwantskap "BEVAT" en 'n verbintenisgraad van EEN : BAIE aan 'n opsionele lidmaatskapsklas entiteit "PERSOON" koppel, d.w.s een kantoor kan baie of geen persone bevat. As die lidmaatskapsklas aan die "BAIE" kant nie-opsioneel is nie, moet daar altyd ten minste een van die "BAIE" gekoppelde entiteite bestaan.

Die UEV-model is geïdentifiseer, nou moet na die toepassing van die model gekyk word, nl. hoe word die model gebruik om 'n probleem op te los ?

4.7 DIE TOEPASSING VAN DIE EV-MODEL

Daar kan sekere stappe gevolg word om 'n EV-model vir 'n werklikheidsprobleem, op te stel. Die proses waarby die EV-model opgestel word, word "entiteitsanalise" genoem en vind plaas onder die vaandel van "data-analise". Davenport [35] [p95] noem dat entiteitsanalise 'n metode verskaf wat gebruik kan word om 'n komplekse omgewing, in terme van entiteite, attributes en verwantskappe te modelleer. Dit is reeds in hoofstuk 2 (fase I) bespreek, alhoewel fase I se toepassing van entiteitsanalise op 'n veel hoër vlak gedoen is. Daar moet altyd in gedagte gehou word dat entiteitsanalise te doen het met die elemente binne die omgewing, en nie met detail oor hoe om die elemente te gebruik nie.

Davenport [35] [p95-96] bespreek sewe stappe wat gevolg moet word, om entiteitsanalise te doen en sodoende 'n EV-model op te stel. Die stappe is soos volg:

- STAP 1: Definieer data-areas en toepassings-areas. Dit kan bv. produksiebeheer, verkope, voorraadbeheer ens. insluit.
- STAP 2: Identifiseer entiteite en entiteit-tipes in elke area nl. plekke, objekte, mense ens. Let op dat die konsep "PLEK" genoem is, wat in die konteks van hierdie verhandeling goed inpas (verspreiding).
- STAP 3: Identifiseer die verwantskappe en grade tussen die entiteite. Stel dit diagrammadies in 'n EV-model voor.
- STAP 4: Identifiseer en beskryf die attribuut-tipes van elke entiteit-tipe bv. naam, adres, nommer, ouderdom, ens.
- STAP 5: Gebruik Codd [3] se normalisasietegnieke om die attributes te ondersoek en funksionele afhanklikhede

aan te toon.

STAP 6: Herhaal stappe 2 tot 5 vir alle toepassingsareas, sodat die verskillende EV-modelle per toepassings-area saamgevat kan word in 'n omvattende EV-model.

STAP 7: Die funksionele EV-modelle en die omvattende EV-model moet vergelyk word om seker te maak dat niks verlore gegaan het nie, of bygevoeg kan word nie. Ons het dus 'n organisasiewye EV-model geskep wat die werking van die verspreide organisasie sal kan beskryf.

Daar moet opgelet word dat ons reeds 'n mate van entiteits-analise in vorige fases van hierdie metodologie voorgestel het, waarin Davenport se stappe 1 en 2 gedeelteliks gedoen is. Verdere stappe word in fase III (huidige hoofstuk) gedoen, sodat 'n EV-model vir 'n verspreide organisasie opgestel word. Davenport se stappe is hoofsaaklik van toepassing op die oorspronklike EV-model. Ons gaan nou kyk na stappe wat van toepassing op die UEV-model is.

Teorey, Yang en Fry [31] [p198-199] identifiseer drie basiese stappe in 'n metodologie vir die opstel van 'n uitgebreide EV-model en relasiemodel. Daar moet opgelet word dat hierdie stappe elemente bevat wat verder as die opstel van die suiwer EV-model strek. Die drie stappe is:

STAP 1: Stel 'n uitgebreide EV-model van die behoeftes op, deur die data te analiseer en te modelleer, sodat semantiese strukture vir opsionele verwantskappe, drieledige verwantskappe en subtipes ingesluit word. Toepassingsbehoefte moet gedefinieer word met natuurlike taal. Spesifiseer ook die uitvoeringsfrekwensie van die toepassings. Kombineer die afsonderlike logiese modelle tot een globale model. Die uitset van hierdie stap is 'n globale UEV-model.

- STAP 2:** Transformeer die globale UEV-model na relasies. Verwyder oorbodige relasies. Die uitset van hierdie stap is kandidaat-relasies (potensiële relasies).
- STAP 3:** Normaliseer die kandidaat-relasies. Identifiseer funksionele afhanklikhede en multi-waarde funksionele afhanklikhede om die sleutels van die entiteite te identifiseer. Funksionele afhanklikhede tussen sleutels moet gevind word deur om na die behoeftes te kyk. Normaliseer die relasies tot die hoogste moontlike graad deur om standaard normalisasietegnieke te gebruik. Die uitset van hierdie stap is genormaliseerde relasies.
- STAP 4:** Plaas die relasies op 'n relasie-DBBS. Hierdie stap behels dus die fisiese implementering van die databasis.

Teorey, Yang en Fry [31] (p199) noem dat hierdie stappe binne die metodologie wat hulle gedefinieer het, werklike voordele vir die ontwerp van baie groot databasisse inhou, omdat dit 'n ekstra konseptualiserings-stap bysit wat die aantal data afhanklikhede wat ontleed moet word, verminder. Daarom is die gebruik van die UEV-model geregverdig vir verspreide databasisontwerp.

Die metodologie wat deur Teorey, Yang en Fry voorgestel is, word gebruik as basis vir die ontwerp van 'n verspreide databasis alhoewel elemente van Davenport se stappe ook ingesluit word. Die voordeel van Teorey, Yang en Fry se metodologie berus op die feit dat dit 'n fisiese implementeerbare struktuur skep. So 'n fisiese implementeerbare struktuur (relasies) is noodsaaklik vir die verdere ontwerp van 'n verspreide databasis, omdat verdere stappe in die ontwerp van 'n verspreide databasis, fragmentasie en replisering van 'n relasies insluit.

Dit blyk dus om 'n goeie keuse te wees, om Teorey, Yang en Fry se metodologie as basis vir die ontwerp van 'n verspreide EV-model en relasiemodel te gebruik. Daar word ook getoon dat transformasie tussen die EV-model en die relasiemodel moontlik is.

Die metodologiese raamwerk vir verspreide databasisontwerp wat in hierdie verhandeling voorgestel is, gebruik hoofsaaklik 'n sintese van tegnieke wat deur verskillende persone voorgestel is. Die eienskap wat hierdie metodologiese raamwerk uniek identifiseer, is gesetel in die verspreide aard van die data, waarop die metodiek van toepassing is. Ons gebruik die basiese teorie van data-analise om ons ontwerp te doen, alhoewel verskeie ander dimensies soos verspreidingsanalise, relasie-generering ens. bygevoeg word om die ontwerper se taak te vergemaklik. Ons kyk volgende na die implikasies betrokke by die opstel van 'n verspreide EV-model.

4.8 'n UITGEBREIDE EV-MODEL VIR VERSPREIDE ORGANISASIES

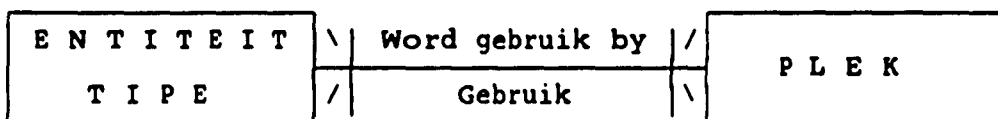
Ons het reeds in die vorige hoofstukke die aspekte betrokke by 'n verspreide organisasie, ontleed. Die een gedagte wat beklemtoon word, kom in die konsep : "PLEK" voor. Ons het dus heertyd te doen met die verskillende "PLEKKE" wat betrokke is by die struktuur van die organisasie, en die entiteite en funksies wat by elk uitgevoer word. Vir die opstel van 'n uitgebreide EV-model vir 'n verspreide organisasie, het ons inligting nodig oor die entiteite van die organisasie, en die entiteite se plekke van voorkoms.

Al die verskillende entiteite (oftewel entiteit-tipes, entiteit-stelle) wat in die organisasie bestaan, kom by 'n spesifieke plek of plekke voor. Hierdie entiteite het deurgaans 'n verbintenis met die entiteit "PLEK". Neem in ag dat ons die uitgebreide EV-model gebruik, waarby die konsep

entiteit-tipe of entiteit-stel nie ingesluit is nie. Teorey, Yang en Fry [31] [p200] gebruik net die term "entiteit" met dieselfde betekenis as die tradisionele EV-model se "entiteit-stel". Daarom noem ons dat verskillende plekke omvat kan word binne 'n entiteit "PLEK" wat verdere verwantskappe met alle ander entiteite bevat.

Dit is dus implisiet aan die aard van 'n verspreide organisasie, dat 'n verwantskap tussen entiteit "PLEK" en alle ander entiteite bestaan. Hierdie verwantskap kan beskryf word as : "BESTAAN BY" of "KOM VOOR BY" met die verskillende verbintenisgrade van 1:1, 1:N, en M:N.

Martin [2] [p321] meld dat vir die opstel van die organisasie-model, word vier objekte gebruik nl. organisasie-eenheid, plek, funksie en entiteit-tipe. Let op dat die objek "PLEK" hierby ingesluit is. Hy gaan voort deur 'n EV-diagram op te stel wat al hierdie objekte as entiteit-tipes beskou. Hy beskou geografiese plekontleding as 'n afsonderlike gedeelte by die opstel van 'n organisasie se struktuurdiagram. Hy toon aan dat objek "entiteit-tipe" aan die objek "plek" verbind is deur 'n M:N verwantskap nl. "WORD GEBRUIK BY" en "GEBRUIK". Dit lyk diagrammaties soos volg:



Die objek "plek" kan dus verbind word met alle ander objekte wat onder objek "entiteit-tipe" voorkom. Dit is dus noodsaaklik om 'n entiteit "plek" binne 'n UEV-diagram van 'n verspreide organisasie in te sluit.

Ons beskou vervolgens 'n benadering wat opgestel is deur die besprekings in die vorige afdelings, en die behoeftes van 'n verspreide en gesentraliseerde databasis te kombineer.

4.9 SINTESE VAN 'n VERSPREIDE EV-MODEL

As gekyk word na die gesentraliseerde geval, is gevind dat die toepassing van die EV-model, verskillende stappe volg, wat hoofsaaklik stapsgewyse verfyning en die opstel van 'n globale EV-model insluit. Die resultaat van die proses wat gevolg word om 'n EV-model vir 'n gesentraliseerde databasis op te stel, is 'n gesentraliseerde globale EV-model. Sekerlik kan die tegnieke wat vir die gesentraliseerde geval gebruik is, effektief gebruik word vir die opstel van 'n EV-model vir 'n verspreide organisasie.

'n Organisasie behoort as 'n geheel beskou te word, sodat die struktuur van die organisasie ontleed kan word en insig daarin verkry kan word, hetsy dit 'n gesentraliseerde organisasie of verspreide organisasie toegepas is. Die opstel van 'n globale EV-model vir die hele organisasie is noodsaaklik.

Omdat 'n gesentraliseerde organisasie 'n gesentraliseerde EV-model nodig het om die struktuur van die organisasie aan te toon, behoort ons 'n verspreide EV-model te gebruik om die struktuur van 'n verspreide organisasie aan te toon. Ons het reeds vooraf verspreidingsanalise in hierdie metodologie voltooi, wat aantoon watter entiteite en funksies bymekaar hoort, en by watter plekke dit voorkom. Hierdie analise kan nou gebruik word as inset tot die verspreide EV-model.

Vir die globale EV-model van 'n verspreide organisasie, kan ons "PLEK" as 'n entiteit aantoon, sodat alle ander entiteite daaraan geskakel word. Hierdie model pas ook baie goed in by die konseptualiseringsaspekte van ontwerp, omdat geen fisiese verwysings na plekke soos Johannesburg, Pretoria ens. gebruik word nie.

Vir 'n verspreide EV-model moet ons die entiteit "PLEK" vervang met die name van die werklike plekke wat voorheen in die metodologiese stappe geïdentifiseer was.

Ons fragmenteer dus die globale EV-model om verspreide EV-modelle oor die verskillende plekke te vorm. Hierdeur sit ons 'n fisiese dimensie by die EV-model wat dit minder van 'n logiese model maak, as wat die globale EV-model is.

Deur dit te doen, toon ons die werklike verspreide struktuur van die organisasie in terme van 'n verspreide EV-model, aan. Regverdiging hiervoor spruit uit die stelling :

" 'n Gesentraliseerde EV-model vir 'n gesentraliseerde organisasie en 'n verspreide EV-model vir 'n verspreide organisasie"

Daar moet ook in gedagte gehou word dat ons eerstens 'n globale EV-model vir die verspreide organisasie ontwikkel, sodat ons 'n beter insig in die struktuur van die organisasie en die verwantskappe tussen die entiteite, in geheel, kan kry. Dit is dus 'n globale konseptuele logiese model.

Daarna vorm ons 'n verspreide EV-model, wat die EV-modelle by elke geografiese plek voorstel. Hierdie benadering, en waar dit inskakel met die vorige stappe van die metodologie, kan soos volg voorgestel word:

STAP 1
Identifiseer poel
van entiteite.

Vind entiteite wat reeds
in Fase I en II ontleed was.

STAP 2
Stel globale EV
model op.

Fase III - stel uitgebreide
globale EV-model op en toon
die entiteit "PLEK" aan.

STAP 3
Stel verspreide EV
model op.

Gebruik Globale EV-model,
verspreidingsanalise en
affiniteitsanalise (Fase II)
om die model te fragmenteer.

GAAN VOORT MET DIE METODOLOGIE.

Die verspreide EV-model kan nou omgeskakel word in relasies, wat beteken dat ons daarna 'n verspreide relasiestruktuur van die databasis sal hê. Die voordele van bg. stappe in die ontwerp van 'n verspreide databasis is:

- 1) 'n Globale EV-model verskaf 'n beter konseptuele beeld van die datastruktuur van die organisasie en word ook gebruik as die bron vir verdere verfyning van die globale model en die verspreide model.
- 2) Ons behou die voordele van bo-na-onder ontwerp en stapsgewyse verfyning in die ontwerpproses.
- 3) Die resultaat is 'n EV-model wat die werklike verspreide struktuur van die organisasie aantoon.

Deur bg. stappe te gebruik, ontstaan daar 'n teenstrydigheid in die teoretiese definisie van 'n logiese model. a.g.v. die feit dat ons 'n fisiese dimensie soos plekname in die verspreide EV-model byvoeg. Dit is egter nie so in die globale EV-model nie. Hierdie teenstrydigheid moet aanvaar word as deel van die unieke eienskappe van 'n verspreide organisasie en 'n verspreide EV-model.

Gedetailleerde ontwerpstappe wat in besonder gevolg moet word, om 'n UEV-model op te stel, word deur Teorey, Yang en Fry [31] voorgestel. Ons het voorheen reeds in vorige gedeeltes, na aspekte van hulle uitgebreide EV-model gekyk.

Die uitgebreide EV-model kan opgestel word deur sekere riglyne te volg om sodoende 'n sistematiese benadering tot ontwerp te gebruik. Ons het reeds na die drie stappe gekyk wat deur Teorey, Yang en Fry [31] (p198-199) in 'n metodologie voorgestel is.

Binne fase 1 van hul metodologie volg hulle die volgende stappe, (p204) vir die opstel van die uitgebreide EV-model: 1) Klassifiseer entiteite en attributes, 2) Identifiseer die

veralgemenings-hierargië en substel- hierargië, 3) Definieer die verwantskappe tussen die entiteite, 4) Integreer die resultate binne 'n globale uitgebreide EV-model.

Die ontwerpstappe kan gebruik word om die globale uitgebreide EV-model op te stel. Die gebruik van 'n uitgebreide model gekoppel aan die stappe, verskaf ook 'n goeie benadering vir die oorskakeling na 'n relasie-model. In terme van 'n verspreide uitgebreide EV-model, moet ons 'n verdere stap bysit nadat die globale uitgebreide EV-model ontwikkel is nl. Stap 5) die opstel van verspreide EV-modelle.

In hierdie stap word die uitsette van fase II van die ontwerpmetodologie vir die ontwerp van 'n verspreide databasis, gebruik. In die opstel van die globale uitgebreide EV-model sal die entiteit "PLEK" moet voorkom, omdat dit inherent aan die aard van 'n verspreide organisasie is.

Hierdie entiteit moet 'n fisiese dimensie aanneem om werklike plekke voor te stel, bv. Johannesburg, Pretoria en Kaapstad. Die uitgebreide EV-modelle van hierdie plekke sal uit die globale uitgebreide EV-model geskep word, met die verskil dat die entiteit "PLEK" nie daarin tevoorskyn sal kom nie. Omdat ons vanaf 'n globale na 'n verspreide EV-model oorgegaan het, is ons daarvan verseker dat die totale datastruktuur van die organisasie ontleed was vir die opstel van die verspreide uitgebreide EV-model.

Palmer en Davenport [12] meld dat data op twee verskillende maniere versprei kan word nl. 1) 'n opgedeelde databasis en 2) 'n gerepliseerde databasis. Opgedeelde databasisse ontstaan waar die konseptuele databasis in gedeeltes verdeel word, wat aan verskeie verwerkers, (plekke) toegeken word. Gerepliseerde databasisse ontstaan waar gedeeltes, of die hele databasis by die verskillende verwerkers of plekke voorkom. As gevolg van hierdie twee moontlikhede, is dit noodsaaklik om 'n verspreide EV-model en 'n globale EV-model vir die databasis op te stel.

Die volgende stap wat op die verspreide UEV-model en die globale UEV-model uitgevoer kan word, is die oorskakeling daarvan na 'n relasiemodel. Die fisiese oorskakelingsproses gaan in 'n verdere fase van die verspreide databasis ontwerpmetodologie gedoen word (Fase V - Logies-fisiese relasiemodellering).

Die teorie en stand van navorsing i.v.m. omskakeling vanaf die EV-model na die relasiemodel word volgende behandel.

4.10 OMSKAKELING VANAF 'n EV-MODEL NA 'n RELASIEMODEL

Soos voorheen genoem, het Teorey, Yang en Fry [31] 'n metodologie vir die ontwerp van 'n relasiedatabasis voorgestel, wat 'n UEV-model skep en omskakel na 'n relasiemodel. Hulle bespreek stappe wat gebruik kan word om 'n reeds opgestelde UEV-model om te skakel na 'n relasiemodel, [p199] wat ook normalisasie insluit. Die toepassing van hierdie stappe gaan in hoofstuk 6 gedoen word waar ons fase V (Relasiemodelling) bespreek. Die moontlikheid van omskakeling is in hierdie hoofstuk genoem, (fase III) omdat regverdiging verkry is uit die feit dat die bron vir omskakeling na 'n relasiestruktuur altyd 'n EV-model is.

Die basiese stappe wat deur Teorey, Yang en Fry [31] [p215] vir die omskakeling van 'n EV-model na 'n relasiemodel genoem word, sluit die volgende in:

STAP 1: Transformeer elke entiteit na 'n relasie wat die identifiserende- en beskrywende-attributes van die entiteit bevat.

STAP 2: Transformeer elke M:N een- of twee-ledige verwantskap na 'n relasie wat die identifiserende attributes van die entiteit/e en die beskrywende attributes van die verwantskap bevat.

STAP 3: Transformeer die drieledige (of n-ledige) verwantskappe na relasies, deur vasgestelde reëls te gebruik.

STAP 4: Die funksionele afhanklikhede tussen entiteite behoort in bg. stappe behoue te bly, maar verdere normalisasie behoort gedoen te word, om te verseker dat die relasies genormaliseer is.

'n Bespreking van bg. stappe kan in Teorey, Yang en Fry [31] (p215) se artikel gevind word. Ons verwys ook verder na ander voorgestelde transformasiemetodes, wat op die EV-model gebaseer is.

'n Baie belangrike eienskap van die EV-benadering is dat dit gebruik kan word vir heen-en-weer oorskakeling na verskillende logiese databasismodelle. So vind ons dat 'n netwerk- en hiërargiesemodel oorskakel kan word na 'n EV-model en die EV-model verder oorgeskakel kan word na 'n relasiemodel (Hwang en Dayal [23] (p235)). Sodoende kan die EV-benadering se voordele as transformasiemodel, ten volle benut word. Die EV-model kan gebruik word om 'n standaard koppelvlak tussen verskillende modelle te definieer.

Hierdie aspek kan in besonder handig te pas kom waar bestaande sentrale databasisse, wat verskillende logiesefisiese modelle gebruik, oorgeskakel word na 'n EV-model wat dan na 'n relasiemodel getransformeer kan word. Die gebruikers se subskemas kan ook gestandardiseer word sonder om dit in terme van ander spesifieke modelle voor te stel. Selfs oorskakeling vanaf 'n CODASYL model na ander modelle kan met die hulp van die EV-model gedoen word.

Hwang en Dayal [23] (p237) toon prosedures aan, wat gevolg kan word om vanaf 'n CODASYL model na die EV-model te transformeer en dan die EV-model na 'n relasiemodel te transformeer deur algoritmes te gebruik. Die rede vir die gebruik van 'n intermediaire model (EV-model) is gegrond in die feit dat

direkte transformasie tussen die netwerk of hiërargiese modelle en die relasiemodel moeilik is, omdat verskillende DMT's (Datamanipuleertale) gebruik word.

Hwang en Dayal [p239] definieer 'n EV gebaseerde taal nl. ERL, wat as basis vir die oor en weer omskakelings tussen die CODASYL en EV-model kan dien. ERL word gebruik om die databasis-skema in EV-terme te definieer deur sekere reëls te gebruik. Hulle gebruik dan 'n algoritme nl. TTrans om die ekwivalente relasiemodel vir die EV-model te skep. Die algoritme TTRANS wat deur Hwang en Dayal [23] [p242] gebruik word om van 'n tradisionele EV-model, na 'n relasiemodel te transformeer, bevat die volgende stappe:

STAP 1: Vir elke entiteit-tipe moet 'n relasieskema geskep word wat al die attributes van die entiteit-tipe bevat. Die sleutels van die entiteit-tipes word die sleutels van die relasies.

STAP 2: Vir elke swak entiteit-tipe moet 'n relasieskema geskep word wat die swak entiteit-tipe se attributes en primêre sleutel bevat, insluitend die attributes wat deel uitmaak van die sterk entiteit-tipe. Die swak entiteit-tipe se primêre sleutels moet die sleutel van die relasie word. Die swak verwantskap-tipe, vanaf die sterk na die swak entiteit, word nie gebruik nie, omdat die sterk entiteit se sleutel in die relasie voorkom.

STAP 3: Definieer 'n relasieskema vir elke verwantskap-tipe met die sleutel-attributes van die entiteit-tipes wat verbind word. Die relasie se sleutels hang van die tipe verwantskap af nl. 1:N of M:N.

Ons sien dat bg. stappe eenvoudiger is as die stappe wat deur Teorey, Yang en Fry [31] [p215] bespreek word. Die rede hiervoor is dat Hwang en Dayal nie 'n uitgebreide EV-model gebruik as bron vir die transformasie nie. Hulle bespreek net

dat so 'n transformasie wel moontlik is. Verder bespreek Hwang en Dayal [23] {p243-251} metodes wat gebruik kan word om relasie-navrae, om te skakel na EV-navrae wat weereens oorgeskakel kan word na netwerk-hiërargiese navrae. Hulle beskryf dus metodes wat gebruik kan word om herwaarts en derwaarts tussen verskillende modelle te navigeer, deur die EV-benadering as tussenstap te gebruik. Dit onderskryf ook die gebruik van die EV-model as 'n konseptuele model vir heterogene verspreide databasisse.

Dumpala en Arora [25] {p337} bespreek ook die moontlikheid van transformasie vanaf 'n EV-skema na ander modelle naamlik die netwerk-, relasie- en hiërargiese modelle (die term 'skema' kan vervang word met die term 'model'). Hulle verskaf ook 'n algoritme wat gebruik kan word om vanaf 'n relasieskema na 'n EV-skema te transformeer. Terugwaardse transformasie is ook moontlik.

Dogac en Chen [27] {p357} toon 'n verfynde oorskakelingsproses omdat hulle meen dat semantiese betekenis verlore mag gaan met growwer oorskakelingsprosesse. Die praktiese ontwerpsondervinding wat hulle opgedoen het deur die EV-model, vir 'n eksperimentele DBMS nl. METUGDBMS, te gebruik, het ook getoon dat die relasies wat verkry was, maklik en sonder probleme op konvensionele datamodelle toegepas kan word.

Om seker te maak dat verwantskappe behoue bly by in oorskakeling vanaf die EV-model na 'n relasiedatabasis, noem hulle die gebruik van 'n "sneller"-konstruk wat sal verseker dat integriteit behoue sal bly. So 'n sneller bevat inligting oor wanner ekstra prosesse op 'n databasis uitgevoer moet word om bv. te verseker dat alle meegaande relasies (swak relasies) saam met 'n eenaar-relasie verwyder word. Hierdie snellers word saam met die relasies voorgestel, sodat hulle altyd uitgevoer kan word met die gebruik van daardie relasies.

Bg. bespreking beteken dat alle ontwerp en analise, in terme van 'n EV-model gedoen kan word, en op basis enige ander

bestaande logies-fisiese databasis-model, geïmplementeer kan word.

As verdere ondersteuning van bg. feit noem Setzer en Lapyda [24] (p319) dat 'n EV-model met min moeite na 'n ADABAS model oorgedra kan word. 'n Moontlike probleem is gegrond in die bewys, dat die gereduseerde model se relasies, wel in 3NF sal verskyn. Hulle meen ook dat verdere navorsing in die outomatisasie van so 'n proses gedoen moet word.

Geoutomatiseerde oorskakeling na enige logies-fisiese skema vanaf 'n EV-model, kan 'n baie handig hulpmiddel t.o.v. verspreide databasisontwerp wees, omdat dit 'n groot gedeelte van die menslike foute in modeloorskakeling kan uitskakel. Ons het in bg. bespreking gevind dat oorskakeling tussen die EV-model en die relasiemodel moontlik is. Dit is dus net verdere ondersteuning vir die keuse van beide die EV-model en relasiemodel, as modelle vir die ontwerp van verspreide databasisse.

4.11 OPSOMMING

Deur die vorige gedeeltes saam te vat, vind ons eerstens dat die keuse van die entiteitverwantskapmodel, as 'n model vir die ontwerp van verspreide databasisse, geregverdig is. Tweedens is dit 'n goeie keuse om die EV-model saam met die relasiemodel te gebruik omdat verskeie skrywers dit moontlik gevind het, om vanaf die EV-model na die relasiemodel te transformeer.

As spesifieke EV-model, kies ons die uitgebreide EV-model van Theory, Yang en Fry [31] omdat dit reeds goeie konstruksie bevat vir EV-modelering en stapsgewyse oorskakeling na die relasiemodel. Ons het ook gevind dat 'n globale EV-model en verspreide EV-modelle ontwikkel moet word om die "PLEK"-entiteit (logies en fisies), te dek. Daar moet besin word

oor die stappe wat gevolg moet word, om die globale uitgebreide EV-model en die verspreide uitgebreide EV-modelle op te stel.

Daar is gevind dat die entiteitgebonde en plekgebonde inligting wat in die vorige fases ingewin was, handig te pas kom as inset tot hierdie fase. Vir verdere ontwikkeling van hierdie fase na 'n volledige metodiek, moet die stappe wat gevolg word in die analise en opstel van die UEV-modelle, gespesifiseer word. Die uitset van hierdie fase is 'n globale UEV-model en verspreide UEV-model, wat met min moeite oorgeskakel kan word na 'n relasiemodel. Hierdie hele fase, sal doeltreffend binne bv. 'n CASE implementering, (Computer Aided Software Engineering) geoutomatiseer kan word.

HOOFSTUK 5

FASE IV : LOGIESE TOEPASSING 3PESIFISERING:

Opsomming

In hierdie hoofstuk (Fase IV) gaan ons die voorafgaande funksionele ontleding van die organisasie gebruik, om die toepassings wat op die databasis uitgevoer moet word, te spesifiseer. Hierdie toepassings kan as prosesse, wat ontaard in rekenaarprogramme of menslike aksies, beskou word. Ons gebruik hoofsaaklik die bo-na-onder benadering vir die identifisering van die toepassings, d.w.s tradisionele stelselontledingskonsepte. Die analise van organisatoriese funksies, (FASE I en II) word gebruik as invoerbron vir die spesifisering van die toepassings. Ons gaan 'n prosesmodel skep waarmee die funksionaliteit van die stelsel gedefinieer kan word. Die prosesse kan bv. in terme van standaard EV-gebaseere datamanipuleerstellings gedefinieer word, of 'n aanvaarbare databasismanipuleertaal soos SQL, kan gebruik word. Dit is belangrik om 'n goeie analise van die toepassings te doen, omdat dit die sukses van die data-replisering, fragmentasie en verspreide databasisprestasie kan beïnvloed. Die uitvoer van hierdie fase, is 'n dokument wat die logiese toepassings definieer, in terme van verspreide gerekenariseerde en menslike prosesse. Ons beskou ook die prosesse in terme van 'n dinamiese model.

5.1 INLEIDING

Normaalweg sal 'n analise van die toepassings nie in databasisontwerp gedoen word nie, omdat alle aandag aan die ontwerp van die datastrukture gespandeer word, sodat daar verseker kan word dat bykans enige toepassings op 'n latere

stadium gedefinieer kan word. Die idee van 'n databasis is dus om 'n datastruktuur te skep waarop enige toepassing op enige tydstip op die databasis gedefinieer kan word. Vir die implementering van verspreide databasisse, is dit wel nodig om die belangriker toepassings vooraf te spesifiseer.

Ceri en Pelagatti [1] (p69) is van mening dat die volgende i.v.m. toepassings en die ontwerp van verspreide databasisse, waar is:

"Although the design of application programs is made after the design of schemata, the knowledge of application requirements influences schema design, since schemata must be able to support applications efficiently. Thus, in the design of a distributed database, sufficiently precise knowledge of application requirements is needed."

Hulle meen dat dit nodig is om inligting i.v.m. die belangriker toepassings in te win, nl. die toepassings wat baie uitgevoer gaan word, of die toepassings wat kritieke prestasiebehoeftes het. Om te verseker dat die belangriker toepassings geïdentifiseer is, moet die minder belangrike toepassings ook geïdentifiseer word, om 'n vollediger beeld van die toepassings te vorm. Die afsnypunt tussen belangriker en minder belangrike toepassings, kan dan meer effektief bepaal word, omdat daar metingspolariteit bestaan wat die belangrikheid van die toepassings vir die databasis sal kan beskryf.

Champine [15] (p135) meen dat verdeling (partitionering) van toepassings baie belangrik is by verspreide databasisontwerp. Verdeling van toepassings is die opdeel van globale toepassings met baie internode kommunikasie, na kleiner verspreide toepassings, sodat relatief min kommunikasie tussen die kleiner toepassings voorkom. Toepassing spesifisering het dus ook ten doel om kommunikasie tussen toepassings te minimaliseer.

Die volgende gedeelte handel oor die konsep van logiese toepassing spesifisering vir verspreide databasisse.

5.2 LOGIESE TOEPASSING SPESIFIKASIE VIR VERSPREIDE DATABASISONTWERP

Vir die doeleindes van verspreide databasisontwerp, moet die belangriker toepassings wat op die verspreide databasis uitgevoer gaan word, geïdentifiseer word. Dit moet op 'n logiese vlak, d.w.s 'n beskrywende vlak onafhanklik van fisiese rekenaartale gedoen word. Dit word gedoen omdat toepassing spesifikasie in 'n latere fase van die ontwerpsmetodologie (hoofstuk 6) gebruik gaan word, om die relasies te dupliseer en te fragmenteer. 'n Goeie beeld van die gebruik van die databasis, en die toepassings wat daarop uitgevoer gaan word, word ook gevorm.

Die prestasie van die toepassings sal die sukses en die aanvaarding van die verspreide databasis beïnvloed. Onthou dat datakommunikasie 'n tyd-kritieke faktor in die implementering van 'n verspreide databasis is. Ons moet in gedagte hou dat stadige toepassings, d.w.s toepassings wat heelwat afstandstoegang (remote access) doen, die gebruikers kan frustreer as die toepassing vinniger uitgevoer behoort te word. Daar is altyd ook alternatiewe databasis-strukture moontlik wat 'n toepassing se uitvoerspoed kan verhoog.

Daar moet ook besef word dat die prestasie-aspekte van toepassings op 'n verspreide databasis streng onderhewig is aan voordeel/nadeel-afspeling in die ontwerp van die verspreide databasis. Hierdie voordeel/nadeel-afspeling kan geoptimaliseer word deur 'n goeie begrip van die toepassings te vorm, sodat daar optimale verspreide databasisontwerp gedoen kan word.

Ceri en Pelagatti [1] (p69) is van mening dat die volgende inligting i.v.m. die verspreide databasis toepassings ingewin moet word ni:

1. Die plek of node waar die toepassing uitgevoer (issued) gaan word. Dit is dus die toepassing se oorsprongsnode.
2. Die aktiveringsfrekwensie van 'n toepassing d.w.s. die frekwensie van aktivering van elke toepassing by elke plek, (node) per tydseenheid.
3. Die hoeveelheid, tipe en statistiese verspreiding van toegang na data deur elke toepassing wat op elke data-objek uitgevoer is, d.w.s toegangsanalise.

Dit kom dus daarop neer dat 'n analise van die toepassings wat op die data-objekte uitgevoer gaan word, gaan plaasvind. Die term "proses" gaan gebruik word om aksies te definieer wat nie noodwendig enige verwysing na rekenaars sal bevat nie, terwyl die term "toepassing" spesifiek vir rekenaar gebaseerde prosesse gebruik gaan word.

Om die prosesse wat op 'n databasis uitgevoer word, te vind, moet ons die data-analise van die organisasie neem, (in vorige fases gedoen) en dit verder ontleed. In die vorige fases van die metodologie is daar data-objekte, funksionele-areas, funksies, entiteite, verspreidingsanalise en 'n globale en verspreide EV-model opgestel. Hierdie inligting gaan gebruik word om die databasis prosesse te identifiseer.

Die volgende gedeeltes gaan aspekte van proses-en toepassing-spesifikasie ontleed. Ons begin deur op die funksionele-analise aspekte van data-analise te konsentreer.

5.3 VERDERE FUNKSIONELE-ANALISE OP VERSPREIDE DATABASISSE

Ons het in die vorige hoofstukke wel aspekte van funksionele-analise bespreek. Funksionele-analise kan nou verder gevoer word, totdat redelike laë vlak definisies, van prosesse en toepassings, gevind word. Tot op hierdie stadium het ons inligting ingewin i.v.m. die funksies wat by verskillende plekke of nodes van die verspreide databasis uitgevoer kan word.

Ons kan die globale funksies by elke plek van die databasis ontleed, verder opbreek en fyner definieer. Ons weet ook watter data-objekte van toepassing is omdat ons reeds data-analise in terme van die verspreidheid van die data-objekte gedoen het (Fase II). Die belangrikste inligting i.v.m. funksies is plek gebonde. Ons behoort dus iteratiewe funksionele-analise van al die funksies by al die plekke of nodes te doen.

Wat die uitvoering van funksies in die verspreide konteks betref, kan 'n funksie of proses drie tipes toegang tot data verlang nl. 1) net toegang tot lokale data, 2) net toegang tot verspreide data d.w.s data by ander nodes en 3) gemengde toegang tot data d.w.s toegang tot lokale en verspreide data in verskillende verhoudings. Ons kan dus iteratiewe funksionele-analise op hierdie inligting toepas, deur elkeen van bg. tipes toeganklikhede afsonderlik te hanteer. Die funksionele-analise gaan grootliks op die EV-model gebaseer word, omdat die EV-model 'n konseptuele-logiese beeld van die data toon. Onthou ook dat so 'n model nie noodwendig 'n refleksie van 'n rekenaardatabasis is nie, maar ontwikkel is deur die organisasie as geheel in gedagte te hou. Ons kan dus nie die gevolgtrekking maak dat die resultate van die funksionele-analise na rekenaar gebaseerde prosesse sal ontaard nie, maar wel organisatoriese prosesse sal definieer.

'n Mate van globale funksionele-analise is reeds in die vorige fases van die ontwerpmetodologie gedoen, in hierdie fase moet

die analise verder gevoer word, sodat die prosesse en toepassings op 'n redelike laë vlak gespesifiseer kan word en die eienskappe van elke proses gedokumenteer kan word. Die stappe wat hiervoor gevolg moet word, sal in die volgende gedeeltes bespreek word.

5.3.1 FUNKSIONELE-ANALISE VAN VERSPREIDE PROSESSE

Aspekte van verdere funksionele-analise moet ontleed word, sodat metodiese stappe gevolg kan word om funksionele-analise verder te neem, (as wat in fase I en II die geval was) en om funksionele-analise suksesvol te voltooi. In die vorige fases is 'n matriks gevorm wat funksionele-analise bevat.

Martin [2] [p358] meen dat so 'n matriks die entiteite teenoor die funksies moet voorstel en elke moontlike funksie / entiteit interaksies moet toon, saam met die tipe toegang wat die funksie na die entiteit het. Die tipes toegang wat gespesifiseer is, (Martin [2]) is skep (C - Create), skrap (D - Delete), lees (E - Read) en bywerking (U - Update). 'n Funksie/entiteits-matriks behoort opgestel te word wat die tipe toegang sal reflekteer. So 'n matriks lyk bv. soos volg:

		FUNKSIES			
ENTITEITE	Funksie 1	Funksie 2	Funksie n	
Entiteit 1	R	R		RU	
Entiteit 2		CRUD			
Entiteit 3	RU	RU			
Entiteit 4				CRUD	
Entiteit 5	R			RU	
Entiteit n		CRUD		R	

Ander moontlike probleme soos entiteite wat nie geskep word nie, of nooit gelees word nie, kan ook in so 'n matriks uitgewys word. Daar kan bv. ook funksies wees wat geen entiteite gebruik nie. 'n Entiteit/Funksie-matriks is dus 'n hulpmiddel wat gebruik kan word om die interaksie tussen funksies en entiteite aan te toon.

Let op dat daar nog nie enige verwysing na verspreiding is nie. Bg. matriks behoort vir die verspreide databasis uitgebrei te word om die tipes verspreide toegang tot die entiteite te reflekteer nl. lokale toegang, verspreide toegang en gemengde toegang. Dit kan opgestel word deur die verspreide UEV-model vir elke plek te gebruik, en die tipe toegang te spesifiseer deur die moontlike toegangsaksies en toegangsbehoefte van elke proses te analiseer.

In fase II (Hoofstuk 3) is verspreidingsanalise gedoen, wat die plekke waar die funksies en entiteite voorkom, getoon het. Ons het dus 'n goeie beeld opgestel, van die verspreide aard van die organisasie in terme van organisatoriese entiteite en funksies (prosesse).

In fase III (hoofstuk 4) is daar 'n verspreide UEV-model geskep, wat die data-struktuur by die verskillende plekke aantoon. Om hierby aan te sluit, moet die funksies verdeel word in prosesse wat dan op die verspreide UEV-model toegepas kan word.

Ons gebruik dus die globale UEV-model om die konseptuele verspreide beeld van die prosesse te reflekteer, d.w.s die verspreide prosesse van die hele organisasie word op die globale UEV-model toegepas. Die gebruik van die globale UEV-model verskaf dus 'n ander benadering tot funksionele-analise omdat dit 'n geheelbeeld van die prosesse wat uitgevoer word, verskaf. Dit word dan later na 'n verspreide beeld oorgeskakel, deur die verspreide UEV-modelle te gebruik.

Martin [2] (p328) meen dat funksies in prosesse opgedeel moet word. Die funksies wat in die vorige stappe van funksionele-analise geïdentifiseer is, moet dus nou in prosesse opgedeel word. Dit word vir globale funksies en verspreide funksies, wat dan in globale of verspreide prosesse sal ontaard, gedoen. Onthou dat die inligting inherent aan 'n funksie, oorgedra word na die prosesse waaruit die funksie bestaan. Die volgende toon die eienskappe van funksies en prosesse (Martin [2] (p329)):

Funksies: Dit beskryf 'n groep aktiwiteite wat een aspek van die progressiewe missie van die organisasie identifiseer. 'n Funksie is aaneenlopend en kan nie onderbreek word nie. Dit beskryf wat gedoen word, en nie hoe dit gedoen word nie, en is gewoonlik 'n woord soos bv. Advertering, Verskeping, Vervaardiging, Arbeidsverhoudinge, Finansiering ens.

Prosesse: Beskryf 'n spesifieke aktiwiteit wat deurgaans in die organisasie uitgevoer word. Dit kan in terme van invoere en uitvoere beskryf word, en dit bevat 'n spesifieke begin en einde. Dit definieer wat om te doen, en nie hoe dit gedoen word nie. 'n Prosesbeskrywing bevat gewoonlik handelings-beskrywings soos bv. Versamel strokies, Bestel onderdele, Vervaardig masjien.

Let op dat prosesse nie rekenaargerigte terme bevat nie, d.w.s dit kan deur 'n mens of masjien gedoen word. Databasis toepassings moet aspekte beskryf wat op die toegang en manipulasie van verspreide databasis entiteite, gerig sal wees. Die spesifikasie van die toepassings sal gebruik word om die fragmentering en replisering van die relasies in Fase V te doen.

Die proses van verdere funksionele-analise omvat 'n groot mate van tradisionele stelselontleding en programmatuur ingenieurswese omdat soortgelyke bo-na-onder en stapsgewyse verfynings gevolg word om die resultate te kry.

Daar is ook verskeie hulpmiddels wat ons kan gebruik om die laë vlak funksionele spesifikasie mee te doen, soos bv. 'n EV-model gebaseerde datamanipulasietaal of relasiemodel gebaseerde taal soos SQL. Die volgende gedeelte bespreek EV-model gebaseerde tale, wat as hulpmiddel vir die logiese toepassing spesifisering van verspreide databasisse gebruik kan word.

5.3.2

ENTITEITVERWANTSKAP GEBASEERDE TALE VIR DIE SPESIFISERING VAN VERPREIDE DATABASIS TOEPASSINGS

As ons 'n EV-model as konseptuele model gebruik, is dit wenslik om die logiese prosesse wat daarop uitgevoer word, in terme van 'n EV-gebaseerde taal te spesifiseer. Sodoende kan 'n gedeelte van die toepassings in dieselfde terme as die van die konseptuele model gespesifiseer word. Die gebruikers kan funksionele aspekte van die EV-model, in terme van konseptuele logiese toepassings spesifiseer, wat op die konseptuele logiese EV-model, toepasbaar is.

Die moontlikheid bestaan selfs om 'n fisies implementeerbare EV gebaseerde taal vir verspreide databasisse te ontwikkel. Vir die doeleindes van hierdie bespreking word die klem geplaas op die gebruik van EV-gebaseerde tale vir die spesifisering van logiese toepassings op verspreide entiteitverwantskapsmodelle.

Lusk en Overbeek [44] (p445) bespreek die moontlikheid van 'n datamanipuleertaal vir entiteitverwantskaps-

modelle. Hulle noem dat so 'n taal op 'n logiese model toepasbaar moet wees, en 'n optimale oplossing moet kan verskaf. Hierdie taal moet as navraag-en spesifikasietaal vir belangrike toepassings gebruik kan word. Die taal wat hulle voorgestel het, is toepasbaar op 'n uitgebreide EV-model. Verder kan die taal ook omgeskakel word na bestaande databasismanipuleertale soos die van IMS, IDMS en selfs SQL.

Die taal wat deur Lusk en Overbeek [44] voorgestel word, [p456] bevat datamanipulasiestrukture en gevorderde strukture wat by fisiese implementasies van 'n EV-model sal kan aanpas. Die volgende gedeelte beskryf die basiese bevele wat uitgevoer kan word en bevat verwysings na veranderlikes (v) en databasisnodes (node).

Die DECLARE (v1,node stel),(v2,node stel)... stelling word gebruik om veranderlikes en die meegaande nodestelle te verklaar. RETRIEVE word gebruik om herwinning van data te doen deur gekombineer te word met stellings NEXT, FIRST, LAST, ENTITY, KEYED-ACCESS, HOLD ens. Verdere stellings soos RELEASE, REPLACE, DELETE, RESET en INSERT word gebruik om verdere datamanipulasie uit te voer. 'n Verdere voordeel wat 'n "generiese" EV-gebaseerde taal inhou, is die moontlikheid om omgeskakel te word na ander manipulasietale.

Ons behoort 'n EV-gebaseerde taal, as 'n logiese konseptuele taal, ten volle te benut. Onthou dat fisiese aspekte soos benodig deur databasis-fragmentasie, gespesifiseer moet word, alhoewel dit op 'n latere vlak gedoen word.

Wat die verspreide entiteitsverwantskapsmodel betref, behoort die taal aangepas word om ook logiese plekke in te sluit. Dit is moontlik om so 'n taal te ontwikkel deur aandag aan die behoeftes van verspreide navrae soos, bv. die lokaliteit van die data, te skenk.

Goldman en Wile [45] [p413] bespreek 'n taal wat gebruik kan word om prosesse te spesifiseer, deur gebruik te maak van 'n verwantskap gebaseerde datastruktuur. Hulle noem dat hierdie "verwantskap" struktuur soortgelyk is aan Chen se EV-model, omdat die verwantskappe tussen objekte (entiteite) gespesifiseer kan word. Objekte word vooraf verklaar of gedefinieer, deur dit in terme van data-tipes te spesifiseer bv.

```
type plane, airport, terminal, cargo, crewmember, route  
end type
```

Name kan nou verder as lede van die data-tipes gespesifiseer word soos bv.

```
type airbus, boeing747, a kind of plane;  
      pilot, stewardess, a kind of crewmember;  
end type
```

Verdere objekdefinisies omvat grense, en die moontlike waardes wat 'n objek kan aanneem bv. JHB, PTA, KPS, BFT ens. wat plekname in 'n verspreide EV-model kan beskryf. Laëvlak-tipes soos stringe, integrale, getalle, ens. kan ook gebruik word vir spesifikasie doeleindes.

Verwantskappe tussen die objekte kan gespesifiseer word om die rol van die objekte teenoor mekaar, duidelik te definieer. Dit lyk bv. soos volg:

```
relation  
      FLIGHTROUTE( plane,route);  
      LOADING(cargo,terminal);  
end relation
```

Hierby kan sleuteldefinisies gevoeg word. Verdere permutasies stel die ontwerper in staat om die realistiese werklikheid tot 'n groot mate te simuleer bv. in terme van 1:1, 1:N of M:N verwantskappe en naamwoorde

vir die objekte. Beperkings tussen objekte kan ook gedefinieer word bv. "Airbus" vliegtuie kan nie by lughawe "Matjiesfontein" land nie. Bg. stellings is deel van die datadefinisie aspekte van die taal. Prosesdinamika kan gespesifiseer word deur objekte en data te skep, te verwyder en by te werk met stellings soos create, destroy, insert, delete en update. Verder kan tradisionele strukture soos begin en end vir groepering, if-then-else vir besluitneming, en whenever-do vir iterasies gebruik word.

Daar kan ook aksies (action, end action) gedefinieer word wat as prosedures kan dien deur stellings saam te groepeer. 'n Spesiale tipe aksie (demon, end demon) kan gedefinieer word, wat uitgevoer kan word, sodra 'n sekere voorwaarde of predikaat geldig word bv. 'n aksie "stormwaarskuwing" sal uitgevoer word wanneer die objek "stormstatus" die waarde "kritiek" aanneem.

Alhoewel Goldman en Wile se taalstruktuur op beide die relasie- en EV-modelle van toepassing is, verskaf dit 'n natuurlike hulpmiddel vir die spesifisering van prosesse en werklike gebeurtenisse. Hulle betree die velde van objek-gebaseerde tale en gebruiker gerigte vierde generasie taalstrukture. Logiese toepassing spesifikasie d.m.v. so 'n taal kan dus werklike voordele vir die ontwerper van 'n verspreide databasis inhou.

Wat verspreide datastrukture en modelle betref, sal die verspreidingskonsepte, beperkings en verspreidingsienskappe outomaties gereflekteer moet word, omdat werklike gebeure noukeurig gespesifiseer moet word. Verwysings na logiese en/of fisiese plekke, moet as deel van die taal se definisie ingebou en gebruik word. Ons moet dus poog om 'n logiese datamanipuleertaal vir verspreide EV-modelle te ontwerp.

Deur bg. taalaspekte te verfyn en aan te pas om as 'n praktiese hulpmiddel vir die spesifisering van logiese toepassings te gebruik, kan die resulterende taal doeltreffend gebruik word in 'n fase vir die ontwerp van verspreide toepassings. Die logiese en konseptuele aspekte van so 'n taal is veral belangrik, omdat dit as 'n modelleringsmeganisme gebruik kan word.

Ons moet verseker dat die toepassings wat d.m.v. so 'n taal gespesifiseer en ontwerp is, na 'n fisies implementeerbare datadefinisie en datamanipuleertaal omgeskakel kan word, om die huidige stand van databasisbedryfstelsels te komplimenteer. Die ontwikkeling van so 'n taal vir logiese toepassing spesifisering op verspreide databasisse, is belangrik om te verseker dat die toepassings, ten volle ontwerp en gespesifiseer word.

Bg. bespreking is hoofsaaklik gebaseer op konseptuele en logiese strukture. 'n Verspreide databasis verminder die logiese en konseptuele eienskappe van 'n model, omdat die aspek van fisiese "PLEK" daarby ingesluit moet word. Ons verbind die model met 'n werklike fisiese struktuur. Dit is noodsaaklik omdat die eienskappe van 'n verspreide databasis dit regverdig.

Vervolgens word die aspekte betrokke by verdere analise van toepassing op die verspreide EV-model, bespreek.

5.3.3 TOEPASSINGSGERIGTE AFFINITEITSANALISE

Ons het reeds in hoofstuk 3 (fase II) kennis gemaak met affiniteitsanalise en saampasbaarheidsanalise. Ons kan dieselfde analises in hierdie fase uitvoer, om affiniteite en saampasbaarheid tussen die veranderlike van "plek" teenoor "toepassings" uit te werk. Dit sal 'n

optimale beeld van toepassingsgroepering verskaf, om die plekke waar die toepassings uitgevoer word, uit te beeld.

Ons kan ook groeperings en affiniteite van toepassings teenoor entiteite, waarna die toepassings lokale toegang het, bereken, om groeperings en lokale entiteite te vorm, en sodoende lokale toegang tot entiteite of data te optimaliseer. Net so kan ons die analyses doen vir eksterne toegang tot entiteite.

Hierdie analyses sal later met mekaar vergelyk moet word, om 'n optimale toepassings-verspreiding te vind, wat op die behoeftes van die verspreide databasis gebaseer is.

5.4 DIE ONTWERP VAN VERSPREIDE TRANSAKSIEMODELLE

Daar is bestaande stelsels wat die aspekte van transaksie-modelle aanspreek wat op die EV-model gebaseer is, nl. Atzeni, Batini, Lenzerini en Villanelli se INCOD stelsel [28] [p375] oftewel "Interactive CONceptual Design for databases".

Hierin vind ons dat hulle 'n UEV-benadering met sukses gebruik om as geoutomatiseerde (rekenaar implimenterde) hulpmiddel vir die ontwerp van databasisse te dien. INCOD adresseer die konseptuele ontwerp, wat se doel 'n geïntegreerde, implimentasie-onafhanklike spesifikasie van toepassings is. Wat uniek hieraan is, is dat hulle ook transaksiemodellering doen. Transaksiemodellering is veral bruikbaar om probleme in die konseptuele skema te identifiseer bv. waar ekstra skema-konstrukte benodig word om 'n spesifieke toepassing uit te voer. Dit is in verspreide databasis toepassings uiters belangrik, omdat die toepassings 'n invloed op die sukses van die databasis het.

Toepassingsanalise kan ook relatiewe toegangsfrekwensies verskaf, wat bruikbaar is vir die logies-fisiese ontwerp van

die databasis. Deur toepassingsanalise te doen, adresseer ons die dinamiese aspekte van die databasis. Die statiese aspekte sluit die verspreide UEV-model in, terwyl die dinamiese aspekte toepassingsgerig is.

'n Bydrae wat INCOD tot databasisontwerp gemaak het, is die skep en onderhou van 'n transaksie-model. Dit stel die ontwerper in staat om die belangrikste transaksies wat op die databasis van toepassing is, te identifiseer. Dit is uiters belangrik vir die fragmentasie en replisering van die verspreide databasis. Die transaksie-model word bedryf deur van 'n transaksietaal gebruik te maak.

Die feit dat Atzeni, Batini, Lenzerini en Villanelli se INCOD stelsel [28] en Sakai, Kondo en Kawasaki [29] [411] se CSDA (Conceptual Schema Design Aid) stelsel, albei die aspekte betrokke by transaksie-modellering aanspreek, is uiters belangrik vir die ontwerp van 'n verspreide databasis. Hierdie stelsels sal prakties aangepas moet word om die behoeftes van verspreide databasis toepassings te hanteer, maar behoort uiters handig as 'n hulpmiddel vir die analise van die logiese toepassings op verspreide databasisse, te wees.

5.5 OPSOMMING

In hierdie hoofstuk het ons gesien dat dit belangrik is om die logiese toepassings wat uitgevoer kan word op 'n verspreide databasis, te analiseer, omdat dit 'n faktor in die sukses van die databasis is. Die toepassinganalise word gebruik om die databasis te fragmenteer en om seker te maak dat die verspreide databasismodel goed genoeg ontwerp is, om die toepassings te kan hanteer.

Verder kan daar prestasiemodellering, gebaseer op die toepassingsanalise, gedoen word sodat vooruitskattings oor die prestasie van die databasis gedoen kan word. Ons kan dus

vroegtydig uitvind of die verspreide databasis 'n sukses sal wees aldan nie.

Die uitset van hierdie fase sal 'n logiese toepassingsmodel, met spesifikasies vir die tipe toegang tot entiteite, uitvoerfrekwensies, uitvoeringsprioriteite en datamanipuleertaal definisies vir die toepassings, wees. Dit beteken dat die sukses van die verspreide databasis vroegtydig verseker word deur op die volledige ontwerp en die analise van die logiese transaksies op die logiese konsepuele model te konsentreer. Daardeur verseker ons ook dat die globale UEV-model en verspreide UEV-model wel die verspreide toepassings sal kan hanteer.

HOOFSTUK 6

FASE V : LOGIES - FISIESE RELASIEMODELLERING:

Opsomming

Logies-fisiese modellering beskryf die fase in die metodologiese raamwerk vir verspreide databasisontwerp, wat te doen het met die implementering van die EV-modelle op 'n relasie DBBS (Databasisbedryfstelsel). Daarvoor gebruik ons die uitsette van die vorige fases (Fases III en IV). Logies-Fisiese relasiemodellering sluit die transformasie van die uitgebreide EV-model (Fase III) na die fisiese implementeerbare verspreide relasiemodel in. Ons ontwikkel dus 'n logiese model wat apparatuur georienteer is en dus ook as fisiese model gebruik kan word. Ons beskou ook die aspekte betrokke by die fisiese implementering van die verspreide databasis nl. fragmentasie, toewysing, en die replisering van die verspreide databasis. Die plasing en fragmentering kan as die fisiese gedeelte van relasiemodellering, wat afhanklik is van werklike plekke, beskou word. Die verspreide relasiemodel word ontwikkel deur die verspreide UEV-model, deur 'n transformasie-algoritme te stuur. Hierdie proses, asook verdere verfynings wat gedoen moet word, gaan in meer besonderheid beskryf word. Ons kyk ook na twee alternatiewe relasie implementerings nl. die transformasie van die globale UEV-model na relasies en transformasie van die verspreide UEV-model na relasies. Fase V gaan ook die implementering van die databasis vooraf, en is die laaste stap van die ontwerpmetodiek. Die resultaat van hierdie fase is fisies implementeerbare verspreide relasies.

6.1 INLEIDING

In hierdie hoofstuk word die relasiemodel wat deur Codd [3] voorgestel is, ontleed, en vind ons redes waarom dit gebruik moet word vir die implementering van 'n verspreide databasis. Die term "logies-fisiese model" word ook ontleed, sodat daar geen verwarring oor die terminologië wat gebruik word, sal ontstaan nie.

Vir die opstel van die verspreide relasiemodel, word die uitgebreide EV-model wat in hoofstuk 4 (Fase III) ontwikkel is, gebruik. Ons het reeds aangetoon dat verskeie algoritmes gebruik kan word vir transformasie van 'n EV-model na 'n relasiemodel. Sulke algoritmes veroorsaak dat minder dinkwerk benodig is vir die opstel van die relasiemodel, as wat nodig sou gewees het in die geval waar die relasiemodel ook as konseptuele model gebruik is. Uitsette, geneem uit die ander fases in die metodiek, word ook deurgaans gebruik as insette wat sal lei tot die voltooiing van hierdie fase.

Een van die belangrikste aspekte wat in hierdie hoofstuk behandel word, is die ontwerp van fragmentasie en die toewysing van die relasies aan die databasisnodes. Hierdie handelinge is een van die grootste onderskeidende kenmerke van verspreide databasisse.

Verder word die relasietafel wat gevorm is, beskou deur die globale UEV-model en verspreide UEV-model te gebruik vir die opstel van die logies-fisiese relasiedatabasis. Hieruit word 'n gevolgtrekking gemaak, wat aanbevelings sal maak oor die mees doelgerigte oplossing vir die ontwerp van die verspreide relasiemodel.

Die volgende afdeling handel oor die konsep van 'n "logies-fisiese model" wat die toepassing van die relasiemodel in hierdie konsep sal aantoon.

6.2 'n LOGIES-FISIËSE MODEL VIR VERSPREIDE DATABASISSE

Ons het reeds in hoofstuk 4 (Fase III - logiese-konseptuele EV-modellering) na die verskillende begrippe "logies", "konseptueel" en "fisies", gekyk. Ons gaan verder 'n kombinasie van twee van hierdie konsepte ontleed nl. "logies-fisies", en ook redes verskaf waarom dit gebruik word.

Soos reeds genoem kan die relasiemodel, netwerkmodel, hiërargiese-model en EV-model as logiese modelle geklassifiseer word, (Korth en Silberschatz [39] (p6-7)) omdat die eienskappe van elk, 'n ontwerper in staat stel om 'n realistiese beeld in 'n abstrakte maar logiese raamwerk voor te stel. Die term "logiese-fisiese model" word aan die relasiemodel gekoppel om die behoefte aan 'n model vir die fisiese ontwerp van 'n verspreide databasis te bevredig.

Die konsep "fisiese model" verwys normaalweg na die laë vlak apparatuurspesifieke implementeringsaspekte van 'n databasis (Korth en Silberschatz [39] (p8)). In terme van hierdie verhandeling d.w.s. vir die implementering van 'n verspreide databasis, gaan ons nie van bg. definisie gebruik maak nie, maar wel 'n ander konsep aan die term "fisiese model" heg nl: 'n fisies implementeerbare model wat nie beperk word tot spesifieke rekenaarargitekturelemente nie. So 'n fisiese model kan op bedryfstelsel-vlak wel met die apparatuur koppel, alhoewel die gebruiker-koppelvlak geen apparatuur-spesifieke verwysings moet bevat nie, en die model logies onafhanklik van die apparatuur is (i.t.v. wysers, lêers). So 'n beskouing stel ons in staat om 'n verspreide databasis fisies te implementeer wat geen spesifieke verwysings na individuele apparaturelemente het nie, d.w.s. logies maar tog ook fisies omdat dit direk implementeerbaar is.

Sodoende definieer ons 'n logiese-fisiese model wat eienskappe van beide konsepte (logiese en fisiese) sal bevat, wat ideaal implementeerbaar sal wees op verspreide argitektuur vir verspreide databasisse. Die keuse van die relasiemodel sal dus

geregverdig word. Die volgende gedeelte verskaf regverdiging vir die keuse van die relasiemodel as 'n logies-fisiese model vir verspreide databasisse.

6.3 DIE GEBRUIK VAN DIE RELASIEMODEL VIR VERSPREIDE DATABASISSE

Ons moet seker maak dat die keuse van die relasiemodel vir die implementering van 'n verspreide databasis, wel die regte keuse is. Dus moet ons die voordele van die relasiemodel vir verspreide databasisse, ontleed, en ook moontlike probleme in ag neem. Die ontwerp van databasisse is al deur baie persone aangepak wat elkeen van verskeie verskillende metodes en metodieke gebruik maak.

Die teorie van relasiedatabasisontwerp kom reeds 'n lang pad, (+- 18 jaar 1970 tot 1988) alhoewel daar huidiglik steeds baie polimiek daaroor is. Sedert die daarstelling van die meer logieser en eenvoudiger relasiemodel van E. F. Codd, [3] (1970) is daar baie vooruitgang in databasis navorsing gemaak, alhoewel in verskeie verskillende rigtings.

Wat die relasiemodel betref, is die grootste belangrikste eienskap 'n eenvoudige voorstelling van data-verwantskappe in terme van "plat-leërs" (flat files) d.w.s. datastrukture waarin elke rekord 'n soortgelyke hoeveelheid velde het, en in 'n twee dimensionele tabel voorgestel kan word (Cardenas [4] {p99}).

Ten spyte van die polimiek dat suiwer relasiedatabasismodelle nog nie werklik kommersieel implementeerbaar is nie, (1986 artikel in COMPUTERWORLD - "Strained relations: DBMS debate turns bitter" [5]) gaan ons van die relasiemodel gebruik maak vir die implementering van 'n verspreide databasis omdat dit baie voordele inhou.

Codd [6] noem in die inleiding van die boek, "Relational

database systems - Analysis and Comparison", vier redes waarom die relasiemodel uiters geskik is vir verspreide databasisse. Hierdie redes is:

1. Relasiedatabasisse bied eenvoudige, buigbare en kragtige fragmenterings- en samevoegings-eienskappe sodat relasietabelle op verskillende plekke maklik vertikaal of horisontaal gefragmenteer kan word (ander DBBS is vasgestrengel in hulle komplekse interne voorstellings a.g.v. navigasiestrukture, wyserskakelings en ander beperkings).
2. Relasieoperatore bied kragtige en dinamiese relasiekombineringsstrukture sodat verspreide inligting maklik voorgestel kan word (soortgelyke strukture bestaan nie in ander DBBS nie).
3. Die relasiedatabasis-datamanipuleertaal bied meer ekonomiese netwerk versendingsmoontlikhede as wat by ander DBBS gevind word, omdat hulle gewoonlik een rekord op 'n slag versend.
4. Die relasiedatabasis-datamanipuleertaal maak dit vir 'n databasisnode in 'n verspreide netwerk moontlik, om die behoeftes van 'n transaksie te analiseer, en dit in basiese komponente op te breek wat dan na ander nodes versend kan word vir lokale uitvoering.

Codd [6] meld dat die oorgrote meerderheid van navorsing in die verspreide DBBS veld gedoen word deur die relasiemodel as raamwerk te gebruik. Dit is ook gevind dat die gebruik van 'n Relasie-DBBS, programmeerder produktiwiteits-verbetering van 5 tot 20 keer tot gevolg het. Produktiwiteitsvoordele is veral handig in verspreide databasisse, omdat dit gewoonlik reuse databasis implementasies bevat wat baie man-jare neem om te ontwerp en te implementeer.

Die relasiemodel bied ook goeie integreringsmoontlikhede wat databasis-elemente soos die databasisbedryfstelsel, 'n navraagfasiliteit, 'n logiese databasisontwerps-hulpmiddel, 'n toepassingsgenerator, datawoordeboek en verspreiding-beheerstelsel betref, alhoewel huidige stelsels nog nie sulke hoë integrasie toon nie. Databasisintegrasie wat die hele stelsel dek, is uiters belangrik vir verspreide databasisse omdat daar verskeie afsonderlike elemente bestaan wat fisies moet kan saamwerk.

Eenvoudige dataherwinning is ook 'n belangrike en voordelige eienskap van die relasiemodel. Die gebruikers kan dus hulle eie transaksies op die databasis makliker spesifiseer en uitvoer (SQL stellings).

Bg. redes ondersteun die keuse van die relasiemodel vir die implementasie van verspreide databasisse. Dit sal dus onnodig wees om 'n verspreide databasis op 'n ander model soos bv. die netwerkmodel of hiërargiese-model te implementeer.

6.3.1 BESTAAN DAAR SUIWER RELASIEMODELLE?

Die vorige motiverings vir die gebruik van die relasiemodel vir verspreide databasisse, soos deur Codd [6] genoem, is gebaseer op Codd se "suiwer" definisie van die relasiemodel soos uiteengesit in bron [3]. In teenstelling hiermee, heers daar tans heelwat polimiek oor die klassifisering van sogenoemde "relasie"-databasisse in terme van Codd se oorspronklike "suiwer" relasiemodel. Die volgende vraag duik nou op : "Is 'n gegewe relasie-DBBS 'n "suiwer" relasiemodel of net soortgelyk (a look-alike), en as dit net soortgelyk is, kan dit nog gebruik word vir die implementering van 'n verspreide databasis ?"

Hierdie vraag kan beantwoord word deur 'n maatstaf te gebruik om uit te vind of die DBBS werklik die "suiwer" relasiemodel ondersteun. Schmidt en Brodie [6] het in 1983 'n studie onderneem waar 'n maatstaf opgestel is om 'n relasie-DBBS te meet. Veertien verskillende DBBS is teenoor hierdie maatstaf gemeet en gevind dat weinige van die sogenaamde "relasie"-DBBS werklik as 'n "suiwer" relasiemodel beskou kan word.

Twee ander artikels in rekenartydskrifte bespreek ook hierdie aspekte, en het dieselfde bevindings. Gallant [5] in sy artikel "Strained relations: DBMS debate turns bitter" wat in die COMPUTERWORLD voorgekom het, noem dat sekere kommersiële relasie-DBBS ontwerpers hulle eie idees en definisies opstel, en nie streng volgens die tradisionele relasiebeginsels werk nie. Daar is as't ware 'n "oorlog" tussen die akademië en kommersiële DBBS-ontwikkelaars, aan die gang.

Codd noem ook in die artikel dat geen kommersiële produk werklik alle relasiemodelreëls ondersteun nie. Die redes hiervoor, soos deur die ontwikkelaars verskaf, het grootliks te doen met die prestasieaspekte van die produkte. Hulle meen dat 'n "suiwer" relasiemodel-implementasie nie aanvaarbaar kan presteer nie. Uit bg. bespreking is dit duidelik dat daar 'n verskil tussen relasiedatabasisteorie en relasie-DBBS implementasies voorkom, wat moontlik probleme en verwarring vir die finale implementasie van 'n verspreide databasis, kan inhou.

Dit is dus 'n aspek wat in gedagte gehou moet word by die praktiese implementasie van die verspreide relasiemodel wat ontwikkel word. Die gekose produk behoort grootliks (amper 100%) aan Codd se oorspronklike definisie onderhewig te wees om te verseker dat dit die unieke eienskappe van verspreide databasisse se logies-fisiese behoeftes kan bevredig.

Volgens die artikel : "Relatively speaking databases still have a long way to go" [10] word die probleme en oplossings vir bestaande relasie-DBBS as volg beklemtoon (spesifiek van toepassing op IBM se DB2):

1. Alle relasie-DBBS behoort verwysingsintegriteit (referential integrity) te kan hanteer. Dit ontstaan waar relasies verwant is aan mekaar, sodat die data-entiteite nie sonder mekaar kan bestaan nie, en data in albei relasies moet wegval as meegaande gedeeltes van die data geskrap word.
2. Die relasie-DBBS behoort intydse prestasie-monitors, skemadefinisiewoordeboeke en uitgebreide samevoegingsoperatore te bevat.
3. Huidige (1988) verspreide databasisimplementasies is onbetroubaar, en gebruik nie 'n twee-fase-protokol (two phase commit protocol) nie, wat tot korrupte databasisse kan lei. Outomatiese dataherstel nadat nodes afgegaan het, is nog nie geïmplementeer nie, omdat daar nie genoeg data-oorbodigheid by die verskillende nodes bestaan nie. Die huidige produkte bevat ook net enkelnode-bywerking en multinode-navrae.

Ons sien dus dat daar heelwat onsekerheid oor die praktiese implementasie van 'n verspreide relasie-databasis bestaan. Daar word ook tans baie navorsing in verspreide DBBS implementering geloots, wat behoort te verseker dat die oorgroote meerderheid van die bg. probleme binnekort (?) opgelos sal word.

As gevolg van die reeds genoemde polimiek, behoort ons eers slegs net die teoretiese aspekte van die relasiemodel te gebruik, omdat daar nie werklik ander alternatiewe bestaan nie. Ons kan dit illustreer deur aan te haal uit PACTEL korporasie se boek "DISTRIBUTED

DATABASE TECHNOLOGY" [42] (p106) :

"Relational Databases: These give the simplest and possibly the only formalised way of representing the problems associated with DDB (Distributed Databases) at this point in time and are therefore useful for theoretical work. Much of the current work on DDB systems that we viewed is being undertaken using the principles of relational theory."

Ons het ook in die voorafgaande hoofstukke reeds aangevoer vir die gebruik van die relasiemodel in die implementering van verspreide databasisse. Ons bespreek volgende die aspekte wat betrokke is by die logies-fisiese relasiemodellering van 'n verspreide databasis, deur die unieke aspekte van verspreide databasisimplementasies uit te wys.

6.4 'n RAAMWERK VIR VERSPREIDE LOGIES-FISIESE RELASIEDATABASIS-ONTWERP

By gesentraliseerde databasisse vind ons dat ontwerp hoofsaaklik om die ontwerp van die konseptuele skema en die fisiese koppeling met die DBBS, draai. Vir verspreide databasisontwerp ontstaan daar 'n hele aantal ander ontwerpaksies a.g.v. die verspreidingsdimensie wat betrokke is by die ontwerp. 'n Groot gedeelte van die ontwerpaksies wat van toepassing is op beide gesentraliseerde en verspreide databasisontwerp, was vooraf in die vorige hoofstukke in terme van die fases van 'n ontwerpmetodologie behandel.

In fase V (logies-fisiese relasiemodellering) is daar ook ontwerpaksies wat beide op gesentraliseerde en verspreide databasisontwerp van toepassing is. Dit sluit hoofsaaklik die opstel van die relasies deur transformasie algoritmes van die EV-model na die relasiemodel in. Ceri en Pelagatti [1] (p86) noem dat 'n groot gedeelte van die ontwerp van 'n verspreide

databasis dieselfde is as in die gesentraliseerde geval, alhoewel die metodologie in hierdie verhandeling ander insigte beklemtoon, en ander perspektief bied.

Die motiewe van ontwerp behoort altyd maksimale benutting en doelgerigtheid na te streef. Unieke ontwerpsaksies wat bykom a.g.v die verspreiding van data is die volgende (Ceri en Pelagatti [1] (p68)):

1. Ontwerp van fragmentasie : Globale relasies moet in terme van horisontale, vertikale en gemengde fragmente uitgedruk word.
2. Ontwerp van die toewysing van fragmente: Replisering, duplikasie en fisiese toewysing van die fragmente in 1. gevind.

Ons sien dat bg. ontwerpsaksies meer te doen het met die fisiese dimensie van verspreide databasisontwerp, as die logiese aspekte daarvan. Daar moet ook onthou word dat hierdie twee fasette ineengeskakel is, en nie onafhanklik van mekaar kan funksioneer nie (daarom die term "logies-fisies").

Anders as by die ontwerp van 'n gesentraliseerde databasis, is baie van die fisiese ontwerpsaksies van verspreide databasisse onderhewig aan die ontleding van die toepassings wat op die databasis uitgevoer moet word. Toepassingsontleding is reeds in die vorige fase (Fase IV - Logiese toepassing spesifisering) gedoen, en dit is dus nie nodig om dit binne hierdie fase weer aan te pak nie. Die uitsette van fase IV bevat onder andere die volgende, wat belangrik is vir die suksesvolle voltooiing van hierdie fase nl:

1. Oorsprong van uitvoering.
2. Frekwensie van uitvoering.
3. Hoeveelheid, tipe en verspreiding van toegang tot die data.

Die bg. inligting word hoofsaaklik gebruik as insette tot die fragmentasie gerigte ontwerpsaksies.

6.5 'n BESPREKING VAN DIE RELASIEMODEL

Ons het vooraf vlugtig na die struktuur van die relasiemodel gekyk. Ons gaan nou weer daarna kyk, en die eienskappe van die model uitwys. Dit is noodsaaklik dat ons 'n goeie oorsigtelike beeld van die relasiemodel verkry, omdat verdere gedeeltes naarna sal verwys.

Alhoewel die model in 1970 (Codd [3]) ontstaan het, is daar tans nog baie navorsing wat op die model gedoen word. Die praktiese aspekte van die model is tans onderhewig aan standardisasie in terme van die SQL-taal oftewel die navraag- en datamanipuleertaal vir relasie databasisse. Ons het reeds vooraf gesien dat die relasiemodel, 'n goeie keuse is vir die opstel van verspreide databasisse omdat dit sekere voordele inhou. Die model se struktuur word in terme van tweedimensionele tabelle voorgestel, sodat die volgende definisie geldig is (Date [4] [p83]):

Definisie: Gegee 'n versameling stellinge D_1, D_2, \dots, D_n , is R 'n relasie van die n stellinge as dit 'n stel georderde n -talle (tuples) $\langle d_1, d_2, \dots, d_n \rangle$ is, sodat d_1 behoort aan D_1 , d_2 behoort aan D_2 en d_n behoort aan D_n . Stellinge D_1, D_2, \dots, D_n is die domeine van R . Die waarde n is die graad van R .

'n Grafiese tabelvoorstelling van relasies lyk soos volg:

RELASIE NAAM	DOMEIN 1	DOMEIN 2	DOMEIN 3	DOMEIN n
STEL WAARDES	A	1	a	I
	B	2	b	II
	C	3	c	III
	D	4	d	IV
	E	5	e	V
	F	6	f	VI

1 Tal

Vir die bg. tabel is die graad van die relasie gelyk aan n (n domeine), elke ry met data is een tal oftewel een n-tal (Omdat daar n domeine is), die aantal talle is die kardinaliteit d.w.s 6 in bg. relasie. Relasies met 'n graad van 1 is eenledig, relasies met 'n graad van 2 is tweeledig, graad 3 is drieledig, graad n is dus n-ledig. Daar is ander faktore betrokke by relasies nl. attributes, sleutels, relasie-dinamika en relasiebetekenis, (Date [41] (p85-91)) wat vervolgens bespreek word.

6.5.1 ATTRIBUTES

Daar is 'n verskil tussen 'n domein ('n ry) en attributes (kolomme) wat uit 'n domein verkry word. Attributes verwys na die gebruik van 'n domein in 'n relasie. 'n Domein kan ook gedefinieer word as 'n data-tipe wat vooraf gedefinieer kan word, en deur verskillende attributes in verskillende relasies gebruik word. Die attributes kan dus as die veranderlikes van domeintipes beskryf word. 'n Proses nl. "normalisasie" moet op die attributes uitgevoer word om die waardes van die attributes so atomies moontlik te maak d.w.s. die attribuutwaarde moet opgebreek kan word na basiese domeine, bv. 'n attribuut NOMMERSNAAM, wat 'n samevoeging van 'n persoon se ID nommer met sy naam is, kan genormaliseer word na twee attributes nl. PERSOONSNUMMER en PERSOONSNAAM, wat van twee verskillende domeine gebruik maak. Die volgende is 'n voorbeeld wat domeine teenoor attributes illustreer d.m.v. 'n hipotetiese relasietaal:

DEFINIEER		
DOMEIN	OUDERDOM	NUMERIES(2).
DOMEIN	NAAM	ALPHABETIES(30).
DOMEIN	TELEFOON	ALFANUMERIES(12).
DOMEIN	WERKNOMMER	ALFANUMERIES(7).

SKEP

```
RELASIE  SEKRETARESSE
( WERKNEMER : WERKNOMMER,
  NOEMNAAM  : NAAM,
  NAAM      : NAAM,
  HUISTEL   : TELEFOON,
  WERKTEL   : TELEFOON,
  OUDERDOM  : OUDERDOM ).
```

Ons gaan volgende aantoon dat 'n tal in 'n relasie uniek identifiseerbaar kan wees, deur relasiesleutels te gebruik.

6.5.2 RELASIESLEUTELS

In elke relasie is daar gewoonlik een attribuut wat die relasie uniek kan identifiseer, sodat die talle van die relasie ook uniek identifiseerbaar is, bv. 'n attribuut soos IDENTITEIT_NOMMER. So 'n attribuut vorm die primêre sleutel van die relasie. Ons vind ook relasies waar meer as een attribuut gebruik moet word om 'n tal te identifiseer d.w.s geen twee talle van 'n relasie sal ooit dieselfde wees nie. Die primêre sleutel kan dus opgebou word uit 'n versameling attributes van 'n tal, sodat die primêre sleutel die minimum aantal attributes gebruik om die tal uniek te identifiseer. Alle moontlike sleutels wat as alternatiewe sleutels gebruik kan word, (ook uniek) word kandidaat sleutels genoem. Die kandidaat sleutels wat nie as primêre sleutel dien nie, word alternatiewe sleutels genoem. Wat inskakeling met die EV-model betref, kan 'n tal gebruik word om 'n entiteit voor te stel, sodat die kandidaat sleutels die entiteit uniek identifiseer. Die relasie kan dus gesien word as die entiteit-tipe. Om seker te maak dat die entiteite wel sonder probleme in terme van die talle voorgestel kan word, moet sekere integriteitsreëls neergelê word nl. entiteits-integriteit en verwysings-integriteit. Dit kom op die volgende neer :

Entiteits-integriteit: Kan gespesifiseer word deur seker te maak dat geen komponent of attribuut van 'n primêre sleutel ooit 'n nul waarde kan verkry nie. Uit die definisie van 'n entiteit moet die entiteit ten volle identifiseerbaar wees.

Verwysings-integriteit: Verwantskappe tussen relasies kan voorgestel word deur twee attributes op 'n primêre domein vir twee verskillende relasies te definieer. Integriteit moet behou word deur te verseker dat die voorkoms van die sekondêre attribuut in die tweede relasie altyd, of nul is, of 'n waarde aanneem van die attribuut van dieselfde domein van 'n hoof relasie waar die attribuut 'n primêre sleutel is. Daar moet dus verseker word dat die waardes van twee attributes van twee verskillende relasiesubstelle van die hoof relasie, attributewaardes bevat bv. as een relasie **PRODUKTE** se sleutel attribuut produk-kodes bevat, moet 'n produk-kode attribuut in 'n ander relasie bv. **VERKOPE** 'n substel van waardes bevat soos aangedui as die produk-kode attribuut van die **PRODUKTE** relasie. Dit sal nie sin maak as 'n produk-kode **PAK01001** in 'n tal van relasie **VERKOPE** voorkom maar die selfde tal produk kode **PAK01001** vooraf uitgewis was uit relasie **PRODUKTE**. Daar is dus verwysing na 'n produk wat verkoop is en wat se kode nie meer bestaan nie. Die afhanklike attribuut (Produk-kode van relasie **VERKOPE**) word ook 'n vreemde sleutel genoem (foreign key).

6.5.3 RELASIEDINAMIKA EN RELASIEBETEKENIS

Date [41] [p90] verwys na relasiebeteckenis (intensions) en relasiedinamika (extension). Dit toon hoofsaaklik die verskil tussen die definisie van die relasie en die dinamiese aspekte van die talle aan.

Relasiedinamika van 'n relasie omvat die aspekte van die relasie wat met tyd verander, d.w.s veranderinge aan talle deur byvoeging, skrapping en bywerking. Relasie-betekenis omvat die statiese of permanente aspekte van 'n relasie d.w.s die spesifisering van die relasieskema in terme van 'n definisie van alle moontlike relasiedinamika.

Hierdie statiese aspekte hanteer die benamingstrukture (domeine, relasie name en attributes) en integriteits beperkinge wat onder andere die sleutelintegriteit en verwysingsintegriteit moet hanteer. Sleutelbeperkinge bevat definisies van al die primêre sleutelattributes en alternatiewe sleutels, en die attributes betrokke, sodat elke tal uniek kan wees en geen nul sleutels kan voorkom nie. Integriteitsbeperkinge hanteer die indirekte verwysing tussen relasies deur vreemde sleutels te spesifiseer.

Ons het reeds die basiese aspekte van die relasiemodel vir DBBS beskou maar nog nie gekyk na die praktiese implementasie en datamanipulasie betrokke nie. Die volgende gedeeltes behandel die praktiese datamanipulasieaspekte wat betrokke is by verspreide relasiedatabasisse.

6.5.4 MANIPULASIE EN DEFINIEERING VAN RELASIESTRUKTURE

Relasie gebaseerde data, d.w.s data wat in die vorm van tabelle onderhewig aan relasiekonsepte gespesifiseer is, kan d.m.v. 'n manipuleertaal nl. SQL, (eers SQUARE, daarna SEQUEL genoem, en toe SQL) gehanteer word. SQL (Structured Query Language) is 'n grootliks Engels-gebaseerde taal, d.w.s dit behoort groot inslag waar Engels as 'n primêre of tweede taal gebruik word, te vind. Dit is tans selfs onderhewig aan standaarde.

Die taal is nie prosedure gerig nie, maar wel objek gerig d.w.s dit is anders as 'n standaard programmeringstaal soos PASCAL. Verder kan die taal bywerking sowel as herwinning hanteer, wat vanaf die bedryfstelsel of gekombineer met 'n programmeringstaal soos COBOL of C uitgevoer kan word. Die datadefinisie-stellings sowel as manipulasie van die datawoordeboek, kan ook deur SQL gehanteer word. Ons kan dus alle relasiedefinisie en manipulasie-aksies op 'n relasie databasis uitvoer, deur die SQL taal te gebruik.

Codd het verder ook ander manipulasiemetodes gedefinieer nl. relasie-algebra en relasie-kalkulus. Prakties vind ons dat SQL aspekte van beide hanteer. Ons gaan nie relasie-algebra of relasie-kalkulus bespreek nie, omdat dit reeds behandel was in hoofstuk 1. Wat huidiglik in die praktiese implementering van 'n verspreide relasiedatabasis belangrik is, is wel die aspekte betrokke by die SQL taal.

6.5.4.1 DIE SQL TAAL EN VERSPREIDE DATABASISSE

Finkelstein [43] (p53) noem in sy artikel "Lingua Franca for Databases", dat SQL besig is om gestandardiseer te word, wat gebruik kan word as standaard databasis kommunikasiemiddel tussen verspreide relasiedatabasisse. Dit beteken dat 'n organisasie tussen verskillende relasiedatabasisse met verskillende apparatuureenhede kan kommunikeer, deur om standaard SQL te gebruik.

Hy beklemtoon dat SQL 'n sleutelrol kan speel in netwerke met verspreide verwerking. Hierom is SQL uiters geskik as standaard databasistaal vir verspreide databasisse.

IBM het 'n behoefte gehad om 'n nie-proseduregerigte taal te ontwerp wat Codd se relasiemodel kan ondersteun nl.

met seleksie, projeksie, samevoeging, union, verskil, produk deling en interseksie, en het toe voortgegaan om SQL te ontwikkel. Huidiglik bied IBM twee relasiegerigte produkte wat 'n groot impak op toekomstige verspreide databasisse behoort te hê nl. SQL/DS en DB2 wat ook na mikrorekenaars (PS2 - OS/2) oorgedra gaan word (1989).

IBM het SQL gebruik in hulle definisie van SAA (System Application Architecture) se gebruikeroppervlak, kommunikasie, verspreide verwerking en ontwikkelings metodologie wat sal verseker dat SQL verdere regverdiging sal verkry. Die toekoms van relasiedatabasisse en SQL is dus verseker.

Tans is daar 'n hele aantal SQL gebaseerde relasiedatabasisse op die mark verkrygbaar soos bv. ORACLE, INGRES, INFORMIX ens. Ongelukkig is SQL nog nie gestandaardiseer nie, omdat daar heelwat verskillende dialekte in gebruik is, wat standardisasie vergemoeilik.

Daar is tog 'n SQL-standaard wat deur ANSI (American National Standards Institute) opgestel is, alhoewel dit nie 'n baie kragtige definisie is nie. IBM se definisie word beskou as die onoffisiële standaard vir SQL, en bevat handige uitbreidings op die ANSI SQL-standaard.

Die ANSI kommittee is huidiglik (1988) besig om uitbreiding te spesifiseer wat verwysings-integriteit, gebruiker-gedefinieerde-integriteit, globale data-definisies en taal-koppelings sal insluit.

Wat verspreide databasisse betref is Finkelstein [43] [p67] van mening dat relasiedatabasisse (SQL) wel verspreide databasisse goed kan ondersteun omdat data d.m.v. logiese waardes adresseerbaar is, wat veroorsaak dat data enige plek in 'n verspreide netwerk kan voorkom sonder om die SQL bevele te beïnvloed.

Heterogene relasiedatabasisse kan ook aaneengeskakel word in terme van verspreiding, mits 'n SQL standaard vir almal gevolg word. Die ORACLE- en INGRES-DBBS hanteer huidiglik lokale bywerking en verspreide navrae, d.w.s geen tweefase-protokolle is geïmplementeer nie, maar planne vir die toekoms sluit volledige verspreide databasis strukture in. Alhoewel daar tans baie voordeel/nadeel-afspeling is, is die neiging vir die toekoms, wat verspreide databasisse en SQL betref, baie duidelik en voorspelbaar.

Ons gaan nou kyk na die struktuur van SQL in terme van 'n datadefinisietaal en as 'n datamanipuleertaal.

6.5.4.2 DIE STRUKTUUR VAN SQL

Die struktuur van SQL kan in terme van drie funksionele areas uitgedruk word (Finkelstein [43] {p55-61}) nl: 1) Datamanipulasie, 2) Datadefinisie en 3) Databeheer. Prakties sal 'n relasiedatabasis opgestel word deur datadefinisies te spesifiseer, en die toepassings in terme van datamanipulasie en databeheer te definieer. Hierdie funksionele areas se stellings is die volgende:

Datadefinisie: Die SQL datadefinisiestellings kan gebruik word om relasietabelle, indekse en sigte (views) te definieer. Die stellings wat hier gebruik kan word is CREATE, TABLE, INDEX, VIEW, UNIQUE, ALTER, AS en dit kan gekombineer word met attributes, domeine en datamanipuleerstellings.

Datamanipulasie: Die datamanipuleerstellings kan gebruik word vir herwinning, byvoeging, verwydering en bywerking van relasietalle (tuples). Die stellings wat hiervoor gebruik kan word is SELECT, UPDATE, DELETE en INSERT met die SELECT stelling wat

die dominante stelling is. Verder kan sleutelwoorde met die bg. stellings gekombineer word in bv. die volgende algemene vorm (vervang ook UPDATE, DELETE, INSERT i.p.v SELECT):

```
SELECT [DISTINCT] item-lys
FROM tabelle
[WHERE seek-voorwaardes]
[GROUP BY colomme]
[HAVING uitdrukking]
[ORDER BY colomme]
[LIKE patroon]
```

Verder kan logiese werkseenhede gedefinieer word wat uit 'n aantal SQL stellings bestaan, wat in totaliteit uitgevoer moet word om integriteit te verseker (geen gedeeltelike bywerking van 'n transaksie word toegelaat nie).

So 'n groep stellings word gedefinieer deur die stellings na 'n BEGIN te spesifiseer. Daarna word die stelling COMMIT of ROLLBACK uitgevoer om te verseker dat die logiese werkseenheid uitgevoer word, of die vorige toestand herstel word.

Databeheer: Hierdie stellings beheer hoofsaaklik die sekuriteit van toegang tot die databasis en gebruik die stellings GRANT en REVOKE, (Selinger [18] [p235 - p238]) wat die gebruikers se toegang tot SELECT, INSERT, UPDATE, DELETE en WITH CHECK OPTION stellings beheer. Databeheer sal ook gewoonlik deur die databasis-administrateur geïmplementeer word.

Verder bestaan daar 'n stelselkatalogus wat die definisies van die databasis stoor, en wat ook gebruik kan word vir navrae soos, bv. om alle tabelle te vind wat 'n sekere attribuut bevat of uit te vind watter gebruikers sekere voorregte het. Die stelselkatalogus is ook vir verspreide databasisse uiters belangrik, omdat dit verwysings, na data op

ander rekenaars bevat (plekke waar fragmente voorkom).

Hieruit is dit duidelik dat SQL 'n uiters geskikte datadefinisie en datamanipuleertaal vir verspreide relasiedatabasisse is, mits dit in terme van een of ander standaard geïmplementeer is, en reg gebruik word. 'n Verspreide databasis moet met die een of ander doelwit in gedagte, ontwerp word. Hierdie doelwitte sal ook as maatstawwe vir die meting van die sukses van die verspreide databasisontwerp en implementasie, kan dien. Vervolgens kyk ons na 'n paar doelwitte wat aan verspreide databasisse geheg kan word sodat dit metingskriteria vir sukses kan voorstel.

6.6 DOELWITTE VIR DIE ONTWERP VAN VERSPREIDE DATABASISSE

Ons kan onself die vraag afvra : "Watter doelwitte moet opgestel word vir 'n verspreide databasis" ?, d.w.s hoe moet die databasis funksioneer om dit so doelgerig, bruikbaar, effektief en prakties moontlik te maak ? Ceri en Pelagatti [1] (p69-70) noem 'n paar doelwitte wat in die ontwerp van 'n verspreide databasis in ag geneem moet word, om die databasis meer bruikbaar te maak. Hierdie doelwitte is:

1. Verwerkingslokaliteit: Die data moet op 'n manier versprei word, sodat maksimale verwerkingslokaliteit verkry word, d.w.s die data moet so na as moontlik aan die toepassings geplaas word. Data kan dus herwin of gebruik word, in terme van lokale navrae of afgeleë navrae. Om te sien hoe data tussen die prosesse versprei moet word, kan ons soos in fase II 'n affiniteitsanalise van data teenoor toepassings doen, net soos ons die affiniteit van data teenoor plek gemeet het. Ons behoort dus die

beste alternatief tussen die lokale navrae en afgeleë navrae tot 'n data-fragment op 'n sekere plek te vind.

2. **Betroubaarheid en beskikbaarheid van verspreide data:** Daar behoort 'n baie hoë graad van beskikbaarheid tot data vir herwinningsdoeleindes, (read-only) te wees. Dit kan gedoen word deur data te dupliseer by verskillende verspreide nodes. Die data moet ook deurgaans bestendig en betroubaar wees, wat elke gedupliseerde gedeelte betref. Korth en Silberschatz (39) (p408) noem dat data replisering (of duplisering) beskikbaarheid aanhelp, verhoogde parallelisme moontlik maak, maar die oorhoofse koste van bywerking sal verhoog.
3. **Verspreiding van werklading:** Die verskillende verspreide rekenaars moet die werklading op die stelsel kan verdeel, d.w.s 'n besige verwerker kan navrae na 'n minder besige verwerker versend. Hierdie motief moet geïmplementeer word in teenstelling met verwerkingslokaliteit. Daar is dus 'n groot mate van voordeel/nadeel-afspeling betrokke tussen doelwitte 1 en 3.
4. **Stoorspasie, koste en stoorspasie beskikbaarheid:** Die nodes in die netwerk kan verskillende stooreenhede bevat, wat tot gevolg sal hê, dat sekere nodes baie groter databasisse as ander nodes, sal bevat. Daar kan dus gespesialiseerde nodes wees (database machines) wat aan algemene nodes gekoppel word. Onthou dat die koste van kommunikasie, verwerkerdyd, invoer / uitvoer-faktore ens. ook in ag geneem moet word.

Die genoemde doelwitte skep baie interessante en komplekse implementerings moontlikhede omdat ons met verskeie uiteenlopende en soms teenoorgestelde doelwitte

moet werk. Ons moet tog 'n keuse maak tussen bg. doelwitte, sodat die gekose doelwit die meeste gewig en inagneming in die ontwerpsproses kan bevat. Een van die meer praktiese doelwitte wat na aan die werklikheid van verspreiding lê, is die verwerkingslokaliteit wat tot outonome maar geïntegreerde verwerking van 'n verspreide databasis kan lei. Die volgende gedeelte handel oor die opstel van die relasiemodel deur fase III se EV-model/le as inset te gebruik.

6.7 OORSKAKELING VANAF 'n UEV-MODEL NA 'n VERSPREIDE RELASIEMODEL

In 'n vorige fase (Fase III - Konseptuele Logiese EV-modellering) het ons twee UEV-modelle opgestel wat die konseptuele en logiese databasis-struktuur van 'n organisasie, in terme van 'n globale en 'n verspreide EV-model uitdruk. Ons kan ook 'n globale relasiemodel en 'n verspreide relasiemodel opstel, deur bg. twee UEV-modelle na relasiemodelle te transformeer. Hiervoor kan ons een van die transformasie-algoritmes vir EV-model na Relasiemodel gebruik. Die gekose transformasie-algoritme wat gebruik gaan word, is die van Teorey, Yang en Fry [31] (p208-215), omdat hulle uitgebreide EV-model gebruik is, om die konseptuele logiese EV-modelle op te stel. Hierdie transformasie-algoritme is in meer besonderheid in artikel [31], beskryf.

Basiese transformasie het drie tipes relasies tot gevolg nl. 1) Entiteitrelasies met dieselfde inligting as die oorspronklike entiteit, 2) Entiteitrelasies met 'n vreemde sleutel afkomstig van die hoof entiteit en 3) Verwantskaprelasies met die vreemde sleutels van al die entiteite wat daaraan verwant is. Die basiese stappe wat gevolg moet word om vanaf 'n UEV-model na 'n relasiemodel te transformeer, is die volgende:

- STAP 1: Transformeer alle dubbel geskakelde entiteite met binêre verwantskappe. a) Transformeer twee entiteite met een binêre 1:1 verwantskap, b) Transformeer twee entiteite met een binêre 1:N verwantskap, c) Transformeer twee entiteite met een binêre M:N verwantskap.
- STAP 2: Transformeer alle enkele entiteite met eenledige verwantskappe. a) Transformeer 'n entiteit met 1:1 of 1:N verwantskap met homself. b) Transformeer 'n entiteit met M:N verwantskap met homself.
- STAP 3: Transformeer alle n-entiteite met meegaande n-ledige verwantskappe.
- STAP 4: Transformeer veralgemeningshierargië en substel-hierargië.
- STAP 5: Transformeer veelvoudige verwantskappe.
- STAP 6: Transformeer swak entiteite.
- STAP 7: Transformeer aggregasie.

Nadat die omskakeling vanaf die UEV-model na die relasiemodel plaasgevind het, is dit wenslik om normalisasie toe te pas deur die funksionele-afhanklikhede en multiwaarde-afhanklikhede van die relasies te ontleed (Teorey, Yang en Fry [31] [p215]). Dit kan gedoen word deur die primêre funksionele-afhanklikhede in die UEV-model te vind, die kandidaat-relasies te ondersoek vir multiwaarde-afhanklikhede en sekondêre funksionele-afhanklikhede, en om normalisasie van die kandidaat-relasies tot op die hoogste vlak te doen, totdat alle oorbodigheid uitgeskakel word.

Nadat normalisasie gedoen is, moet die relasies tussen die nodes van die databasis versprei word. Die opeenvolgende gedeeltes sal hierdie aspekte bespreek.

Ons het in die bg. gedeelte gesien hoe 'n UEV-model omgeskakel kan word na 'n relasiemodel. Onthou dat ons in fase III, (Konseptuele-Logiese EV-modellering) twee verskillende EV-modelle geskep het nl. die globale UEV-model en 'n verspreide UEV-model. Die vraag ontstaan nou: "Watter van die twee EV-modelle moet na die relasiemodel van 'n verspreide databasis, getransformeer word?"

Die meeste van die bronne i.v.m. verspreide databasis-ontwerp (Ceri en Pelagatti [1] {p68}) noem dat die konseptuele ontwerp in terme van 'n gesentraliseerde model gedoen word, waarna dit versprei word d.m.v. fragmentasie. Dit is dus geregverdig om die globale UEV-model na 'n globale relasiemodel te transformeer.

Aan die ander kant behoort dit nie verkeerd te wees om 'n plek gebonde verspreide EV-model na verspreide relasiemodelle te transformeer nie, omdat daar in werklikheid met 'n fisies verspreide databasis gewerk word. Die tegnieke wat toegepas moet word om fragmentasie ens. te doen, is tot 'n mindere mate nodig, omdat die konseptuele ontwerp reeds verspreidingsaspekte gehanteer het. Een van die belangrikste eienskappe van 'n verspreide relasiemodel, is dat kommersiële relasie gebaseerde verspreide databasisbedryfstelsels nog nie die volledig teorie van verspreiding kan hanteer nie. Ons ontwikkel dus 'n verspreide verwerking stelsel i.p.v. 'n virtuele verspreide databasis, terwyl die aspekte van databasis-deursigtigheid vir die hele netwerk nog geldig moet bly.

'n Ander voordeel wat die generering van 'n verspreide relasiemodel inhou, is dat die ontwerpers van die verspreide databasis, baie meer insig in die verspreidingsaspekte van die databasis vind. Dit kan as 'n tussenstap in die ontwerp van die databasis gesien

word, waarby die ontwerpers insig in die verspreide struktuur van die databasis verkry. Omdat verspreide databasisse ook baie groot kan wees, kan hierdie stap werklike en konkrete voordele vir die ontwerpers inhou. Gross, Jackson, Joyce en Mc Guire (16) (p311) noem dat lokale skemas (van 'n verspreide relasiemodel) nodig is om prestasiesimulasie en toetsing vooraf te doen, om te bepaal of die databasis voldoende sal kan presteer.

Ons kan ook die toepassings wat vooraf gespesifiseer is, (Fase IV) makliker teenoor die verspreide model voorstel, as wat ons dit sou kon doen teenoor die sentrale model. Ons kan ook meer insig in die struktuur van die verspreide toepassings, en die ineenskakeling van die verspreide databasis met die toepassings, verkry. Dit kom daarop neer dat elke node as 'n outonome funksionerende databasisseenheid beskou kan word, waar sekere van die toepassings lokale-databasisverwysings sal uitvoer, en ander toepassings na ander nodes se data sal verwys.

Ons kan dus die verspreide relasiemodel as ondersteunende model vir die analise en ontleding van die databasistoepassings, gebruik, bv. die mate van lokale-teenoor verspreide-verwysing van navrae, bywerkings ens. Die verspreide relasiemodel is dus handig vir die analise van praktiese verspreide databasisbehoefte.

Feitlik altyd is dit waar dat die gebruiker op die einde, die verspreide databasis sal gebruik in terme van 'n globale skema of globale relasiemodel. Hulle sien dus nie die verspreide aspekte van die model nie, alhoewel verspreide verwysings in die definisie van die model ingebou moet word soos, fabrieke word gevind in Johannesburg, Durban en Kaapstad.

Wat die gebruiker aanbetref, is hulle besig met 'n organisasiewye virtuele databasis, wat in terme van 'n logiese relasiemodel uitgedruk is. 'n Globale

relasiemodel is dus deurgaans van toepassing. Ons moet dus verseker dat die globale uitgebreide EV-model na 'n globale relasiemodel getransformeer word, omdat die resulterende model belangrik is vir die gebruikers van die databasis, en verdere implementering van die verspreide databasis.

Die navorsingsbronne wat verspreide databasisontwerp beskryf, gebruik almal 'n globale relasiemodel as bron vir die ontwerp van databasis verspreiding, nl. vir duplisering, fragmentasie en toewysing. Die volgende gedeelte hanteer die aspekte wat hierby betrokke is, deur 'n globale relasiemodel vir die ontwerp van 'n verspreide databasis te gebruik.

6.8 DIE ONTWERP VAN DATABASIS VERSPREIDING

Die ontwerp van databasis verspreiding het te make met die ontwerp van fragmentasie deur bo-na-onder ontwerp te volg, sodat nie-oorvleulende fragmente gevind word, wat toegewys kan word na nodes van die verspreide databasis (Ceri en Pelaratti [1] {p72}).

Korth en Silberschatz [39] {p408} ondersteun hierdie feit en noem dat 'n relasie R hoofsaaklik onderhewig is aan replikasie, fragmentasie en 'n kombinasie van die twee.

Replisering van data sal die beskikbaarheid en parallelisme van data verhoog, terwyl dit groter oorhoofse kostes vir bywerking sal meebring. Replisering van 'n relasie word gevind waar die stelsel verskeie identiese kopieë van die relasie by verskillende nodes onderhou (Selinger [17] {p223}). Die alternatief is om net een kopie van die relasie by een node te stoor. 'n Verspreide databasis implementasie sal bykans altyd 'n groot mate van voordeel/nadeel-afspeling tussen replisering en fragmentasie bevat, om die prestasiebehoefte

van die databasis te kan bevredig.

Ceri en Pelagatti [1] [p72] noem dat data-fragmentasie te make het met die groepering van talle (horisontale-fragmentering) en attributes (vertikale-fragmentering) sodat elke groepering, oor dieselfde eienskappe beskik, om sodoende 'n fragment te vorm. Vervolgens word die aspekte betrokke by databasis-fragmentasie behandel.

6.8.1 DATABASIS-FRAGMENTASIE

'n Relasie r kan opgedeel word in fragmente $r_1, r_2, r_3, \dots, r_n$ sodat die fragmente tesame genoeg inligting bevat om die oorspronklike relasie r te herkonstrueer (Korth en Silberschatz [39] [p409]). Die herkonstruksieproses kan uitgevoer voer deur relasieoperatore (JOIN en UNION) op die fragmente uit te voer. Daar kan horisontale of vertikale fragmentasie op 'n relasie uitgevoer word. Horisontale fragmentasie van 'n relasie r word gedoen deur talle (relasie r_y) van die relasie aan een of meer fragmente toe te ken. Vertikale fragmentasie word op 'n relasie r toegepas deur substelle van die attributes aan fragmente toe te ken. Ons kan ook 'n kombinasie van vertikale en horisontale fragmentasie op 'n relasie toepas.

Die stappe wat gevolg moet word by fragmentasie van relasies, is eerstens die ontwerp van die fragmente en daarna die toewysing van die fragmente (Ceri en Pelagatti [1] [p72]). Ons behandel eerstens die ontwerp van fragmentasie waarna die toewysing van fragmente bespreek word.

6.8.2

DIE ONTWERP VAN FRAGMENTASIE

Vir die doeleindes van die ontwerp van fragmentasie, moet ons kyk na horisontale fragmentasie, vertikale fragmentasie en gemengde fragmentasie. Hierdie aspekte sal in die volgende gedeeltes bespreek word. Elam en Fisher [13] [p116] is van mening dat daar ook wiskundige metodes beskikbaar is, om fragmentasie van verspreide databasisse te doen, wat gebaseer word op wiskundige modelle. Wiskundige metodes kan dus ook gebruik word vir die proses van fragmentasieontwerp, wat die objektiwiteit van die ontwerp sal verhoog. Horisontale fragmentasieontwerp word volgende bespreek.

6.8.2.1

HORISONTALE FRAGMENTASIEONTWERP

Ceri en Pelagatti [1] [p73] tref onderskeid tussen twee tipes horisontale fragmentasies nl. primêre fragmentasie en afgeleide fragmentasie, waar afgeleide fragmentasie in terme van primêre fragmente gedefinieer word. Hou in gedagte dat horisontale fragmentasie te doen het met groeperings van relasie-talle. Ons gaan primêre horisontale fragmentering volgende bespreek, waarna afgeleide horisontale fragmentasie bespreek sal word.

Primêre horisontale fragmente word gedefinieer deurdat seleksies ("SELECT" stelling) op globale relasies gedoen word, sodat elke tal van die globale relasie net in een fragment mag voorkom. Die toepassings voer die "SELECT" stellings op die fragmente uit. Let op dat toepassings 'n rol speel in fragmentasie.

'n Relasie word horisontaal gefragmenteer deur 'n seleksiepredikaat (reël of eienskap) op 'n relasie te definieer, sodat die resulterende fragment deur 'n spesifieke toepassing, sinvol gebruik kan word. Die

doeltreffendheid van die predikaat is van die insig van die ontwerper afhanklik (Ceri en Pelagatti [1] (p74)). Ons kan dus 'n goeie seleksiepredikaat of 'n slegte seleksiepredikaat ontwerp. 'n Stel goeie predikate op 'n relasie, moet minimaal en volledig wees. Die volgende is op so 'n stel predikate van toepassing:

" P is 'n stel van predikate p_1, p_2, \dots, p_n op relasie R sodat die stel $P = \{p_1, p_2, p_3, \dots, p_n\}$ kompleet is as enige twee talle van R in die selfde fragment met dieselfde waarskynlikheid deur enige toepassing verwys word en minimaal is as al die predikate relevant is".

Fragmentasie kan gedoen word deur die volgende algoritme op relasies en transaksies toe te pas (Ceri en Pelagatti [1] (p74)):

1. Identifiseer 'n predikaat p_1 wat die talle van relasie R in twee dele sal verdeel sodat die dele verskillend deur ten minste een toepassing gebruik sal word.
2. Identifiseer 'n predikaat p_i wat ten minste een van die bg. fragmente in twee dele sal verdeel sodat ten minste een toepassing, die dele verskillend kan gebruik.
3. Sit die predikaat p_i in die stel predikate $P = \{p_1, \dots, p_i\}$. Elimineer irrelevante predikate.
4. Herhaal stappe 2 en 3 totdat die stel P volledig is.

Daar moet in ag geneem word dat 'n werklike volledige en uitgebreide stel predikate baie tydsam en duur sal wees om te vind, omdat daar 'n hele aantal verskillende tipes

toepassings op 'n verspreide databasis kan voorkom. Daarom behoort ons net 'n paar van die werklik belangrike toepassings te gebruik, en fragmente met soortgelyke eienskappe nie te gebruik nie.

Afgeleide horisontale fragmente van 'n relasie R is nie afhanklik van die attributes van die relasie nie, maar is wel afhanklik van die horisontale fragmente van ander relasies, wat 'n invloed op die toepassings wat R gebruik, uitoefen. Dit is fragmente wat samevoegings ("JOIN") tussen ander fragmente van ander relasies as fragmentasiekriteria gebruik. As verskeie toepassings die fragmente van verskillende relasies saamvoeg op dieselfde samevoegings predikaat, kan 'n alternatiewe fragment geskep word wat deur die spesifieke toepassings gebruik kan word.

Ons spesifiseer dus alternatiewe fragmentasie predikate wat samevoeging bevat, wat uitgedruk is in terme van primêre fragmentasie predikate. 'n Voorbeeld van horisontale fragmentasieontwerp is:

'n Bank besit drie takke nl. in Johannesburg, Durban en Kaapstad. Vir hulle rekeninge gebruik hulle 'n relasie REKENING(TAK, REKNR, PERSOON, BALANS) wat globaal soos volg kan lyk:

REKENING

TAK	REKNR	PERSOON	BALANS
Johannesburg	11112	Mnr A	10000
Johannesburg	13123	Mnr B	400
Johannesburg	13223	Mnr C	2236
Durban	21224	Mev D	476
Durban	23213	Mev E	2123
Kaapstad	32453	Mev F	4567
Kaapstad	35432	Mev G	3424

Die toepassings word lokaal deur elke tak uitgevoer, sodat transaksies by elke tak gehanteer kan word met die SQL stelling:

```
SELECT PERSOON FROM REKENING
WHERE TAK = (waar ook al)
```

Die predikaat is tak gebonde dus P = (TAK='Johannesburg', TAK='Durban', TAK='Kaapstad') sodat fragmente by elke plek bestaan. Daar is geen oorbodige prdikate soos PERSOON = 'Mnr A' teenwoordig nie.

Ons kyk volgende na vertikale fragmentasieontwerp d.w.s fragmente wat attribuut gebonde, en nie tal gebonde is nie.

6.8.2.2 VERTIKALE FRAGMENTASIEONTWERP

Ceri en Pelagatti [1] (p79) meld dat vertikale fragmentasieontwerp te make het met groeperings van stelle attributes van 'n globale relasie R, sodat toepassings die fragmente op dieselfde wyse kan gebruik. Kriteria vir vertikale fragmentasie is dus weereens toepassingsgebonde, net soos by horisontale fragmentasie. Die vertikale fragmente moet die primêre sleutel van die relasie of 'n tal-identifiseerder bevat, sodat elke tal uniek kan wees.

Vertikale fragmentasie bestaan uit vertikale partisionering en vertikale groepering (clustering). Die doel van vertikale fragmentasie is om fragmente te identifiseer, sodat verskeie toepassings uitgevoer kan word deur slegs een fragment van 'n relasie i.p.v. die hele relasie, te gebruik.

Vertikale partisionering is 'n aspek van vertikale fragmentasie waar geen verdubbeling van attributes voorkom nie, d.w.s enige attributes van die vertikale fragmente van 'n relasie kom net een keer in al die vertikale fragmente voor. Die voordeel verbonde aan

vertikale fragmentasie, ontstaan waar twee toepassings, twee verskillende vertikale fragmente van 'n relasie op twee verskillende plekke gebruik. Die kompleksiteit en moontlikhede vir vertikale fragmentasie is proporsioneel afhanklik aan die aantal attributes en die aantal toepassings. Ons kan die volgende twee benaderings gebruik om attribuut partisionering te verrig nl.

1. Verdelings benadering: Die globale relasie word progressief in fragmente verdeel.
2. Groeperings benadering: Attributes word progressief gegropeer om fragmente te vorm.

Herkonstruksie van die relasie kan gedoen word deur die vertikale fragmente net weer saam te voeg. Hou in gedagte dat die talle van een fragment altyd aan die ooreenstemmende tal van 'n ander fragment van dieselfde relasie moet skakel d.m.v. 'n primêre sleutel of talnommer. Hierom kan ons aflei dat vertikale fragmente altyd dieselfde aantal talle sal besit as wat die relasie sou gehad het.

Vertikale Groepering van relasies laat dieselfde oorvleulende attribuut in verskillende vertikale fragmente toe, sodat die behoeftes van die toepassing die beste bevredig word. Ons moet ook ander voordeel/nadeelafspelings doen soos bv. tussen nut, stoorspasie en alternatiewe vertikale fragmente. Vertikale groepering is veral voordelig vir navraag-intensiewe toepassings omdat dit dan nie nodig is om ander attributes uit ander fragmente of relasies te bekom nie.

Vertikale fragmentasie word dus geregverdig a.g.v. die unieke eienskap van transaksies op verskillende plekke dat nie al die attributes van 'n globale relasie benodig word nie.

6.8.2.3 KOMBINERING VAN VERTIKALE EN HORIZONTALER FRAGMENTASIE

Vertikale en horisontale fragmentasie kan gekombineer word om optimale fragmentasie-ontwerp tot gevolg te hê. Dit kan gedoen word deur bestaande vertikale fragmente weereens horisontaal te fragmenteer, of om bestaande horisontale fragmente verder vertikaal te fragmenteer.

Dit is 'n hoogs iteratiewe ontwerpsproses wat deurgaans verskeie veranderlikes gebruik, en wat tot 'n groot mate van vertikale en horisontale onderverdeling van fragmente en relasies kan lei. Korth en Silberschatz [39] [p413] meld dat gemengde fragmentasie die proses is waarby 'n relasie R , horisontaal of vertikaal in verskillende fragmente r_1, r_2, \dots, r_n verdeel word, waarna die fragmente nogmaals deur dieselfde proses gestuur word.

Ons het vooraf na al die aspekte betrokke by die ontwerp van fragmentasie gekyk, en gaan vervolgens die plasing of toewysing van hierdie fragmente beskou.

6.8.3 DIE TOEWYSING VAN FRAGMENTE

Ceri en Palagatti [1] [p82] meld dat daar reeds baie navorsing in die toewysing van fragmente gedoen is, en dat daar nog steeds baie probleme is, omdat dit so moeilik is om 'n optimale model van benutting vir verspreide databasisse voor te stel. Die primêre doel van die toewysing van fragmente is om die hoeveelheid eksterne toegang van toepassings na 'n fragment te minimaliseer.

Daar is 'n paar kriteria wat gebruik kan word om die toewysing van fragmente te bepaal, bv. deur empiriese maatstawwe soos koste en voordeel te gebruik. Die moontlikheid van replisering of duplikasie van fragmente

veroorzaak bykomende kompleksiteit aan die toewysingsprobleem. Fragment duplisering verhoog die aantal veranderlikes in die toewysingsproses, omdat die graad van duplisering en die moontlikhede van alternatiewe keuses by navraag gerigte toepassings belangrik is. Om bg. kompleksiteit in 'n mate te beheer, noem Ceri en Palagatti [1] (p83) twee toewysingsmetodes wat gebruik kan word :

1. Identifiseer alle nodes waar die voordeel wat verkry word deur 'n kopie van die fragment daar te plaas, groter is as die koste van die plasing, en plaas dan een kopie by al daardie nodes. Sodoende identifiseer ons die nodes wat voordeel uit die plasing verkry.
2. Doen volledige toewysing van ongedupliseerde fragmente, voeg daarna duplisering van fragmente progressief by, as daar voordeel daaruit verkry word. Stop die proses wanneer verder duplisering nie groter nut meebring as met die vorige geval nie.

Ceri en Pelagatti [1] (p84-86) bespreek 'n aantal formules wat gebruik kan word vir optimale toewysing van fragmente, sodat analitiese maatstawwe vir toewysing gevind kan word.

Dit kan handig gebruik word vir die implementering van die verspreide databasis. Onder andere word maatstawwe soos die uitvoeringsfrekwensie van 'n toepassing by 'n node, die aantal herwinningsverwysings van 'n toepassing na 'n fragment, en die aantal bywerkingsverwysings van 'n toepassing op 'n fragment, gespesifiseer.

Om na die ontwerp van 'n verspreide relasiedatabasis terug te keer, noem Korth en Silberschatz [39] (p414) dat die eindresultaat oor sekere eienskappe moet beskik nl.

outonome funksionering en netwerk deursigtigheid in terme van die benaming van data-items, die replisering van data-items, fragmentasie, en die nodes waar fragmentasie en replisering toegepas word. 'n Groot mate van die fisiese ontwerp van die verspreide databasis, is dus deursigtig vir die gebruiker. Die gebruiker is dus nie van al bg. aspekte bewus nie.

As gevolg van die bg. feite is dit noodsaaklik om die globale relasiemodel vir die verspreide databasis behoue te laat bly, en sodoende ook die logiese modelleringsaspekte van die relasiemodel te benut, deur die gebruiker toe te laat om die volledige model, sonder fragmentasie en plasingseienskappe, te laat gebruik.

6.9 OPSOMMING

Hierdie fase ontwikkel relasiedatabasis modelle of skemas. Die relasiemodel word vanaf die UEV-modelle, wat in fase III ontwikkel is, gegenereer. Ons het dus nie nodig om 'n konseptuele relasiestruktuur op te stel nie, omdat dit reeds in terme van 'n UEV-model gedoen is. 'n Globale relasiemodel en 'n verspreide relasiemodel word opgestel.

Die globale relasiemodel dien as 'n oorvleulende skema, wat deur die gebruikers en toepassingsprogrammeerders gebruik word. 'n Verspreide relasiemodel word ook opgestel wat hoofsaaklik gebruik word om aspekte van databasis toepassings in meer detail te spesifiseer, omdat dit totale databasis verspreiding reflekteer. Die toepassings sal gespesifiseer word deur van gestandaardiseerde SQL stellings gebruik te maak. SQL, gekombineer met verspreide relasiedatabasisse, is 'n besondere goeie keuse om verspreide databasisse mee te ontwikkel, omdat dit relatief gestandaardiseerd is, en nie fisiese modelleringsaspekte soos wysers en lêers aanspreek nie. Dit verskaf dus virtuele modelleringsmoontlikhede.

Wat die gebruikers van die verspreide databasis betref, sien hulle nie werklike verspreidheid raak nie, maar wel logiese verwysings na plekke. Wat die ontwerper betref, is die fisiese plasing van die relasies uiters belangrik. Die ontwerper sien dus die plekke waarna versprei moet word, fisies raak, asook die logiese verwysings soos deur die gebruikers gesien word. Vir optimale databasis verspreiding moet die ontwerper, aspekte van relasie duplikasie, fragmentasie en toewysing op die globale model implementeer. Om dit uit te voer, is 'n gedetailleerde kennis van die belangrikste toepassings nodig.

In hierdie fase word 'n globale relasiemodel opgestel, wat die ontwerp van duplisering, horisontale fragmentasie, vertikale fragmentasie, gemengde fragmentasie en die toewysing van relasies bevat. 'n Verspreide relasiemodel wat by elke node gebruik kan word om die toepassings te ontwerp en te implementeer, word ook opgestel. Die verspreide databasis is dus nou reg om geïmplementeer te word, omdat ons alle ontwerpaksies verrig het, wat die behoeftes van verspreide databasisontwerp bevredig.

HOOPSTUK 7

SLOTSOM - 'n METODOLOGIE VIR VERSPREIDE DATABASISONTWERP

Opsomming

In hierdie hoofstuk is die bevindinge en besprekings uit die vorige hoofstukke, vlugtig bespreek. Daar word ook tot 'n finale gevolgtrekking gekom, wat die stand van verspreide databasisontwerp, en moontlike praktiese implimentasie van 'n ontwerpsmetodologie, betref. Die verhandeling word afgesluit met 'n objektiewe oorsig oor die inhoud van die verhandeling, en wat moontlik in die bespreking van die onderwerp, (verspreide databasisontwerp) bereik is.

7.1 INLEIDING

Die vorige hoofstukke bevat heelwat verwysing na verskillende benaderings, metodes en tegnieke, wat deur verskillende outeurs voorgestel is, vir die ontwerp van databasisse. Sommige van hierdie verwysings is van toepassing op gesentraliseerde databasisontwerp, maar daar is deurgaans verwys na die moontlikheid om die teorie op verspreide databasisontwerp toe te pas. Die mees waarskynlike en produktiewe benaderings en teorieë is nagevolg.

Daar is bevind dat daar min volledige metodologieë vir verspreide databasisontwerp bestaan, maar dat 'n teoretiese metodologiese raamwerk ontwikkel kan word deur teoriefragmente saam te voeg, en met 'n waarskynlike metodologie tevoorskyn te kom. So 'n raamwerk is in hierdie verhandeling ontwikkel.

Daar moet ook van die verstandhouding uitgegaan word, dat hierdie verhandeling nie poog om die enigste of beste oplossing vir verspreide databasisontwerp voor te stel nie,

maar om net 'n rigting aan te wys waarin verdere navorsing gedoen kan word. Die leser behoort dus met 'n kritiese oog na die voorgestelde metodologiese raamwerk te kyk, die mees toepasbaarste gedeeltes uit te haal, en verbeteringe en ander teoretiese rigtings voor te stel.

'n Organisasie wat 'n verspreide databasis implementeer, sal dit hoofsaaklik doen omdat die struktuur en die data van die organisasie versprei is. Die databasis sal dus soos 'n handskoen op die struktuur van die organisasie pas, en is ideaal om verspreide data mee te hanteer.

Daar is ook ander redes vir die implementering van 'n verspreide databasis, bv. 'n verspreide databasis is geregverdig waar die bron van die data versprei is, en 'n behoefte vir lokale toegang tot die data bestaan. Ander regverdiging vir verspreide databasisse is: 1) kostevoordele, 2) data oorbodigheid a.g.v. replisering van data by verskillende nodes en 3) dinamiese aanpasbaarheid tot veranderende omstandighede omdat elke databasisnode afsonderlik kan aanpas, gebaseer op lokale behoeftes.

Die veld van verspreide databasisontwerp is nog nuut en sal heelwat tyd nodig hê om te groei en te ontwikkel. Dit sal ook streng gekoppel wees aan meegaande tegnologiese verbeterings.

7.2 SAMEVATTING VAN DIE VOORGESTELDE METODOLOGIESE RAAMWERK

Uit die vorige hoofstukke is gevind, dat dit nodig is om 'n metodologie te ontwikkel, wat gebruik kan word vir die ontwerp van verspreide databasisse. So 'n metodologie behoort elemente van die nuutste teorieë van databasisontwerp te bevat, en die behoeftes van verspreide databasisse te kan bevredig.

Die voorgestelde metodologie gebruik hoofsaaklik 'n bo-na-onder benadering tot ontwerp, omdat bo-na-onder stapsgewyse

verfyning sal verseker dat niks misgekyk of uitgelaat word nie, en alle elemente aangespreek word. Onder-na-bo metodes kan gebruik word om die kruisverwysing tussen die fases van die metodologie te behartig, en om te verseker dat alle ontwerpsbehoefte aangespreek is.

Die metodologie moet nie net die "suiwer" data-aspekte aanspreek nie, maar ook die toepassings op die data hanteer. Die analise van die databasis toepassings is 'n integrale deel van die metodologie omdat dit noodsaaklik vir verspreide databasisontwerp is.

'n Modellerings benadering word gevolg, om te verseker dat die gebruikers van die databasis, ook insig in die ontwerp van die verspreide databasis kan verkry (modelle is meer verstaanbaar). Dit is een van die redes waarom die entiteitsverwantskapsmodel as 'n konseptuele ontwerpmodel, en die relasiemodel as logies-fisiese model, gebruik is.

Die relasiemodel word gebruik as implimenteerbare logiese model, omdat dit die struktuur van 'n verspreide databasis kan hanteer, sonder om aan fisiese apparatuurstrukture verbind te wees. Die entiteitsverwantskapsmodel is ook as transformasiemodel gebruik, met die relasiemodel as die eindresultaat (teikenmodel).

Die metodologiese raamwerk wat gebruik is vir verspreide databasisontwerp, bestaan uit vyf fases:

FASE I : ORGANISASIE ANALISE - BEHOEFTEBEPALING EN ONTLEDING:
In hierdie fase verseker ons dat ons alle moontlike inligting i.v.m. die verspreide organisasie bekom, en dit in terme van 'n besigheidsmodel ontleed. Ons probeer om verbeterings voor te stel, en die groei van die organisasie in die ontwerpproses in te bou, deur te verseker dat ons weet waarheen die organisasie as geheel, in die opeenvolgende jare sal beweeg. Ons verseker dat ons die organisasie verstaan, en dat die organisasie en verspreide databasis saam kan groei in

die toekoms. Daar word ook data-analise gedoen.

FASE II : VERSPREIDINGSANALISE: Hierdie fase hanteer die analise van die geografiese verspreiding van die organisasie, en die verspreide struktuur van die organisasie. Ons verseker hierdeur dat ons weet waar die elemente en eenhede van die organisasie voorkom, en hoe dit gegroepeer moet word. Die data-analise word in terme van fisiese verspreidheid uitgedruk.

FASE III: KONSEPTUELE - LOGIESE ENTITEITSVERWANTSKAP-MODELLERING: Hierdie fase stel 'n konseptuele model van die organisasie op, deur die uitgebreide entiteitverwantskapmodel se strukture te gebruik. Ons stel 'n globale UEV-model en 'n verspreide UEV-model op. Die gebruikers kan ook hierdie modelle verstaan. Ons maak dus seker dat ons oor volledige insig in die globale en verspreide konseptuele en logiese datastruktuur van die organisasie, beskik.

FASE IV : LOGIESE TOEPASSING SPESIFISERING : Hierdie fase word gebruik om die toepassings op die UEV-modelle te spesifiseer, om te verseker dat ons die verspreide struktuur en werking van die toepassings op die verspreide databasis, verstaan. Die toepassingsontleding word ook gebruik om die logies-fisiese model te fragmenteer.

FASE V : LOGIES-FISIESE RELASIEMODELLERING : Hierdie fase word gebruik om 'n fisies implementeerbare verspreide databasismodel te skep, en te verseker dat die model optimaal sal funksioneer. Die funksionering van die verspreide databasis is afhanklik van besluite wat in terme van databasis voordeel/nadeel-afspeling geneem is. Ons ontwikkel 'n globale relasiemodel om insig in die totale struktuur van die databasis te verkry. Die gebruikers verwys ook na hierdie model. 'n Verspreide relasiemodel word ontwikkel om insig in die werking van die lokale nodes van die databasis te verkry. Vir fisiese implementering van die verspreide databasis, word

fragmentasieontwerp en plasingontwerp voltooi. Die verspreide databasis is nou volledig ontwerp, en reg vir implementasie.

Omdat verspreide databasisse gewoonlik aan die behoeftes van 'n baie groot organisasie moet voldoen, moet die ontwerpers baie tyd spandeer om insig in die organisasie te verkry. Hulle probeer dus misluknings uitkakel of minimaliseer, deur op alle verspreide databasis behoeftes en ontwerpaspekte te konsentreer. Die ontwerpers maak ook seker dat hulle nie net 'n verlede-gerigte statiese organisasie as ontledingsbron gebruik nie, maak ook vir die toekoms van die organisasie voorsiening te maak, deur toekoms-gerigte aspekte in die ontwerp in te bou.

Die gebruik van die voorgestelde metodologie vir verspreide databasisontwerp, kan beskryf word as "uitputtende ontwerp" (exhaustive design), sodat die sukses van die implementering van die verspreide databasis, d.m.v. goeie ontwerp verseker is. Daar word dus baie aandag aan die ontwerp en analise van die databasis gespandeer. Ons konsentreer dus op die konsep van : "Sukses van implementering d.m.v. uitputtende ontwerp".

7.3 SLOTSOM

In hierdie verhandeling is daar gepoog om 'n raamwerk vir verspreide databasisontwerp voor te stel, wat die sukses van die implementasie van die databasis sal verseker. Daar is na verskeie aspekte gekyk, en die mees waarskynlikste oplossings vir die probleme van verspreide databasisontwerp, is nagevolg. Dit beteken nie dat dit die enigste rigting wat gevolg kan word voorstel nie, maar dat bestaande rigtings en teorieë wel toegepas kan word met die ontwikkeling van 'n metodologie vir verspreide databasisontwerp.

Die raamwerk vir verspreide databasisontwerp, wat in hierdie

verhandeling voorgestel is, is nie volledig nie, en behoort verder in 'n volledige metodologie ontwikkel te word. Die konsepte wat betrokke is in elke gedeelte, is oorsigtelik maar volledig bespreek, om as globale verwysingsraamwerk te kan dien.

'n Versoek word aan die leser gerig om objektief of selfs subjektief na die voorgestelde oplossing vir verspreide databasisontwerp te kyk, en dit te assimileer en te verwerk ten opsigte van die leser se eie ondervinding en kennisveld. Die hoop word uitgespreek dat elemente van die raamwerk vir verspreide databasisontwerp, wat in hierdie verhandeling voorgestel is, sal lei tot handige en praktiese oplossings vir verspreide databasisontwerp in die realistiese werklikheid (regte wêreld).

--*--

B R O N N E L Y S

- [1] Ceri, S., Pelagatti, G., Distributed databases - Principles and Systems, McGraw-Hill Book Company, 1985.
- [2] Martin, J., Information Engineering - Volume 2: Strategies & Analysis, Savant Research Studies, 1986.
- [3] Codd, E.F., "A Relational model of Data for large Shared databanks", Communications of the ACM, 1970, Vol 13 no 6.
- [4] Cardenas, A.F., Database Management Systems, Allyn and Bacon Inc, 1979.
- [5] Gallant, J., "Strained relations: DBMS debate turns bitter", Computerworld, Jan 13, 1986.
- [6] Schmidt, J.W., Brodie, M.L., (ed.), Relational database systems - Analysis and Comparison, Springer - Verlag Berlin, 1983.
- [7] Freeman, P., "Design Fundamentals", Infotech State of the Art Report - Structured Analysis and Design, Hosier, J., (ed), Infotech Institute, 1978.
- [8] Denning, P.J., "A Hard look at Structured programming", Infotech State of the Art Report - Structured Programming, Bates, D., (ed), Infotech Institute, 1976.
- [9] Tsichritzis, D.C., Lochovsky, F.H., Data Models, Prentice-Hall Inc, 1982.
- [10] Atre, S., "Relatively speaking databases still have a long way to go", Computing SA, vol 7, no 25, June 22, 1987.
- [11] Hubbard, G.U., Computer assisted database design, Van Nostrand Reinhold, 1981.

[12] Palmer, I., Davenport, R.A., "On-Line Databases", Infotech State of the Art Report - Distributed databases", White, C.H. (ed.), Infotech International, 1977.

[13] Elam, J., Fisher, M., "The use of mathematical models in distributed database design", Infotech State of the Art Report - Distributed databases - Volume 2: Invited Papers", White, C.H. (ed.), Infotech International, 1979.

[14] Adiba, M., "Issues in distributed database management systems", Proc Conf on Very Large Databases", Berlin, 1978.

[15] Champine, Infotech State of the Art Report - Distributed databases - Volume 1: Analysis & Biography", White, C.H. (ed.), Infotech International, 1979.

[16] Gross, J.M., Jackson, P.E., Joyce, J., McGuire, F.A., "Distributed database design and administration", DISTRIBUTED DATA BASES, Draffen, I.W. (ed.), Poole, F. (mede red.), Sheffield City Polytechnic, 1980.

[17] Selinger, P.G., "Replicated Data", DISTRIBUTED DATA BASES, Draffen, I.W. (ed.), Poole, F. (mede red.), Sheffield City Polytechnic, 1980.

[18] Selinger, P.G., "Authorisation and Views", DISTRIBUTED DATA BASES, Draffen, I.W. (ed.), Poole, F. (mede red.), Sheffield City Polytechnic, 1980.

[19] Chen, P.P. (ed.), Entity-relational approach to information modeling and analysis, Elsevier Science Publishers B.V., 1983.

[20] Teichroew, D., Germano, F., Luca, S., "Applications of the Entity-Relationship Approach", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.

- [21] Chen, P.P., "A preliminary framework for entity-relationship models", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [22] Chen, P.P., "The entity relationship model: toward a unified view of data", ACM transactions on Database Systems, 1 March 1976.
- [23] Hwang, H., Dayal, U., "Using the Entity-Relationship Model for implimenting multi-model database systems", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [24] Setzer, V. W., Lapyda, R., " Design of Data models for the ADABAS system using the entity-relationship approach", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [25] Dumpela, R.S., Arora, S.K., " Schema translation using the entity-relationship approach ", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [26] Elmasri, R., Cog en Rahini, S., " Notes on DDTs: An Apparatus for experimental research in distributed database management systems", ACM SIGMOD RECORD, Vol. 11, No 4.
- [27] Dogac, A.,Chen, P.P.," Entity relationship model in the ANSI/SPARC framework", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [28] Atzeni, P., Batini, C., Lenzerini, M., Villanelli, F.," INCOD: A system for Conceptual design of data and transactions in the entity-relationship model", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.

- [29] Sakai, H., Kondo, H., Kawasaki, Z., "A development of a Conceptual Schema Design Aid in the Entity-Relationship Model", Entity-relational approach to information modeling and analysis, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [30] Lampson, P.W., Paul, M., Siegert, H.J., Distributed systems - Architecture and Implimentation . An Advanced Course., Springer Verlag, 1981.
- [31] Teorey, T.J., Yang, D., Fry, J.P., "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", ACM Computing Surveys, Vol. 18, No. 2, June 1986.
- [32] Kleinrock, L., Distributed Systems, Computer, Nov 1985.
- [33] Teorey, T.J., Fry, J.P., Logical database design: a practical approach , Infotech State of the Art Report, Database technology, 1978.
- [34] White, C.H., (Redakteur), Infotech State of the Art Report-Distributed Databases- Volume 1 : Analysis and Biography., Infotech International, 1979.
- [35] Davenport, R.A., "Design of Distributed Database Systems", Infotech State of the Art Report - Volume 2 : Invited Papers, White, C.H., (red), Infotech International, 1979.
- [36] Case, A.F. Jr., "Implementing CASE", SYSTEM DEVELOPMENT, Volume 7, Nommer 11, November 1987, Applied Computer Research, 1987.
- [37] Mendelson, H., "ECONOMIES OF SCALE IN COMPUTING: Grosch's Law Revisited", Communications of the ACM, Volume 30 Number 12, December 1987, Association of Computing Machinery, 1987.

- [38] Oxborrow, E., Databases and Database Systems, Concepts and Issues, Chartwell-Bratt, 1986.
- [39] Korth, H.F., Silberschatz, A., Database system concepts, McGraw-Hill, 1986.
- [40] Navathe, S., Cheng, A., "A Methodology for database schema mapping from extended entity relationship model into the hierarchical model.", The Entity Relationship Approach To Software Engineering, Davis, G.C. (reg), North Holland, New York, 1983.
- [41] Date, C.J., An Introduction to Database Systems, Third Edition, Addison-Wesley, 1981.
- [42] PACTEL, Distributed Database Technology, NCC publications, Manchester, 1979.
- [43] Finkelstein, R., "Lingua Franca for Databases", PC Tech Journal, Vol 5 no 12, Desember 1987.
- [44] Lusk, E.L., Overbeek, R.A., "A DML for Entity-Relationship Models", Entity-relational approach to systems analysis and design, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [45] Goldman, N.M., Wile, D.S., "A Relational Data Base Foundation for Process Specification", Entity-relational approach to systems analysis and design, P.P. Chen (ed.), Elsevier Science Publishers B.V., 1983.
- [46] Jensen, R.W., Tonies, C.C., SOFTWARE ENGINEERING, Prentice-Hall Inc, Englewood Cliffs, 1979.
- [47] Yeh, R., Roussopoulos, N., Chang, P., "Database Design: an approach and some issues", Infotech State of the Art Report - Database Technology, 1978.